AFWAL-TR-80-3101
Volume II

MULTI-RATE DIGITAL CONTROL SYSTEMS WITH SIMULATION APPLICATIONS

Volume II: Computer Algorithms

DENNIS G. J. DIDALEUSKY
FLIGHT DYNAMICS LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433

RICHARD F. WHITBECK
SYSTEMS TECHNOLOGY, INC.
HAWTHORNE, CA 90250

SEPTEMBER 1980

TECHNICAL REPORT AFWAL-TR-80-3101 VOL II
Final Report — January 1979-May 1980

FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

DENNIS G. J. DIDALEUSKY, Captain, USAF
Project Engineer
Control Dynamics Branch
Flight Control Division

R. O. ANDERSON, Chief
Control Dynamics Branch
Flight Control Division

FOR THE COMMANDER

ROBERT C. ETTINGER, Colonel, USAF
Chief, Flight Control Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIGC, *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/27 March 1981 — 510

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFWAL-TR-80-3101, Vol. 2
2. GOVT ACCESSION NO.: AD-A097 550
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): MULTI-RATE DIGITAL CONTROL SYSTEMS WITH SIMULATION APPLICATIONS. Volume II. Computer Algorithms.
5. TYPE OF REPORT & PERIOD COVERED: Final Report, 29 Jan 79 - 29 May 80
6. PERFORMING ORG. REPORT NUMBER: STI-TR-1142-1-2
7. AUTHOR(s): R. F. Whitbeck, D. G. Didaleusky
8. CONTRACT OR GRANT NUMBER(s): F33615-79-C-3601
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Systems Technology, Inc., 13766 South Hawthorne Boulevard, Hawthorne, California 90250
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Program Element 61102F, Project 2304, Mathematics, Task N3, Math. of Flt. Control
11. CONTROLLING OFFICE NAME AND ADDRESS: Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433
12. REPORT DATE: Sep[redacted]80
13. NUMBER OF PAGES: 122
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):
18. SUPPLEMENTARY NOTES:
19. KEY WORDS (Continue on reverse side if necessary and identify by block number):

Digital Control Systems	Frequency Response	Computational Delays	Servo Analysis
Sampled Data	Switch Decomposition	Tustin Transform	Closed-Loop Analysis
z-Transforms	Closed-Loop Systems	Simulation Error Analysis	
w-Domain	Linear Systems		
Multi-Rate Sampling	Root Locus		

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The report is organized in three volumes. Volume I contains the theoretical developments as well as illustrative examples and case studies. Volume III contains FORTRAN listings for the two computer algorithms described in Volume II.

The two computer programs presented in Volume II provide some of the basic digital implementation tools required in the analysis and synthesis of hybrid systems. The DISCRET computer program converts a general analog or continuous

(continued)

340425

20. ABSTRACT (Continued)

model expressed in the s-plane to the z-, w-, or w'-plane. DISCRET can calculate the standard, delayed, or advanced discrete transform. Data holds including the zero order, first order, second order, and slewer can be inserted into the transformation. The second program, TXCONV, implements the conversion of a high-rate discrete transform to a low-rate discrete transform.

The transform conversion expressions mechanized in TXCONV are developed. These expressions allow a high-rate transform in the z-, w-, or w'-plane to be converted to a low-rate discrete transform. The fundamental definition of the z-transform and the discrete inversion integral form the basis for the analytical development. The computer mechanization is based on the practical calculation of the residues of a complex integral. The residues for this integral are calculated in an unconventional manner using a limiting process via L'Hôpital's rule. This method simplifies the mechanization scheme and leads to a closed-form solution.

Detailed descriptions of the DISCRET and TXCONV computer programs are provided. This includes the available program options, theoretical basis for the mechanization algorithms, general and detailed program structure, required input data, and typical program output.

FOREWORD

The research described in this report was performed by Systems Technology, Inc., Hawthorne, California, under Air Force Contract F33615-79-C-3601. The Task Number N3, Mathematics of Flight Control, was under Project Number 2304, Mathematics. This work was directed by the Control Dynamics Branch, Flight Control Division, Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Paterson Air Force Base, Ohio. The work was administered by Captain Dennis G. J. Didaleusky.

Richard F. Whitbeck was the Systems Technology, Inc., Project Engineer under the direction of Duane McRuer.

The authors wish to express their appreciation to the Systems Technology publication staff for their efforts in preparing this three-volume report.

The authors also wish to express their thanks to Ms. Susan Riedel at Systems Technology, Inc., and to Captain Stanley Larimer and Dr. Robert Schwanz at the Flight Dynamics Laboratory for their appreciable efforts in reviewing the technical report.

This report is organized in three volumes. Volume I contains the theoretical developments as well as illustrative examples and case studies. Volume II describes two algorithms useful in the analysis of multi-rate systems, the DISCRET and TXCONV computer programs. Volume III contains the FORTRAN listings for these computer programs.

This report covers work performed from January 1979 through May 1980. The report was submitted by the authors in August 1980.

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
II.	REVIEW OF FUNDAMENTAL SAMPLED-DATA THEORY	4
	A. Introduction	4
	B. Fundamental Sampled-Data Relationships	4
	C z-Transform and the Inversion Integral	11
III.	DEVELOPMENT OF TRANSFORM CONVERSION EXPRESSION	14
	A. Introduction	14
	B. Transform Conversion in z-Domain	14
	C. Transform Conversion in w- and w′-Plane	18
	D. General Multi-Rate Configurations	20
IV.	DESCRIPTION OF DISCRET COMPUTER PROGRAM	26
	A. Introduction	26
	B. Program Options	27
	C. Mathematical Description of Program	28
	D. Program Structure	32
	E. Description of Subroutines	34
	F. Program Variables Within Labeled COMMON	46
	G. Program Operating Instructions	49
V.	DESCRIPTION OF TXCONV COMPUTER PROGRAM	77
	A. Introduction	77
	B. Program Options	78
	C. Transformation Expressions	80
	D. General Residue Calculation	82
	E. Mechanization Scheme	86
	F. Root Conversion Between s, z, w, or w′ Planes	89
	G. Program Structure	94
	H. Description of Subroutines	96
	I. Program Operating Instructions	99
VI.	SUMMARY	111
REFERENCES		114

LIST OF FIGURES

		Page
1.	General Structure of DISCRET and TXCONV Computer Programs	2
2.	Amplitude Modulation of Impulse Train by Continuous Signal	7
3.	The Complex p-Plane	9
4.	Fast-Input/Slow-Output Sampling with Phantom T/N Output Sampler	15
5.	Fundamental Fast-Input/Slow-Output Multi-Rate Sampling	22
6.	Fast-Input/Slow-Output Sampling with Common Sampling Period T	22
7.	Fast-Input/Slow-Output Sampling with Phantom Sampler T_*	23
8.	Fast-Input/Slow-Output Sampling with Discrete System Component	24
9.	General Form of Sampled System	26
10.	DISCRET Program Structure	33
11.	Sampled Continuous System	49
12.	Sampled Systems for Table 4 Input Data	54
13.	DISCRET Output for Data Set 1	56
14.	DISCRET Output for Data Set 2	59
15.	DISCRET Output for Data Set 3	62
16.	DISCRET Output for Data Set 4	65
17.	DISCRET Output for Data Set 5	67
18.	DISCRET Output for Data Set 6	69
19.	DISCRET Output for Data Set 7	71
20.	DISCRET Output for Data Set 8	73

Page

21. DISCRET Output for Data Set 9.............................. 75

22. High-Rate Input/Low-Rate Output Sampling..................... 78

23. TXCONV Program Structure.................................. 95

24. Fast-Input/Slow-Output Sampled System...................... 101

25. Phantom Sampler Concept.................................. 103

26. TXCONV z-Plane Output for Example 1....................... 108

27. TXCONV w′-Plane Output for Example 2...................... 109

LIST OF TABLES

1. Data Holds Implemented in DISCRET.......................... 29

2. Input Data for DISCRET.................................... 50

3. Alphanumeric Codes for DISCRET............................ 51

4. Examples of Input Data for DISCRET........................ 55

SECTION I

INTRODUCTION

The guidance and control field has traditionally focused on continuous or analog control systems represented in the Laplace or s-domain or in a state-space model. Today, the increasing use and popularity of the digital computer as a system component has provided the major impetus to the theoretical as well as the practical interest in sampled-data or discrete control systems. The basic problem facing the control engineer is obtaining valid discrete models of complex, closed-loop hybrid systems (i.e., systems containing both analog and discrete elements). These discrete models must be in a convenient form that can be readily analyzed using the analysis and synthesis tools available today. Valid discrete models for hybrid systems can be obtained using the z-, w-, or w′-transform. The z-transform is a logical extension of the Laplace transform and can be used to handle sampled-data systems. The w- and w′-transforms are related to the z-transform through simple bilinear transformations. Discrete models expressed in the z-, w-, or w′-plane define the continuous variables in a hybrid system at the sampling instants and completely describe the inherently discrete variables associated with digital elements.

The two computer programs presented in this volume provide some of the basic digital implementation tools required in the analysis and synthesis of hybrid systems. The DISCRET computer program converts a general analog or continuous model expressed in the s-plane to the z-, w-, or w′-plane. DISCRET can calculate the standard, delayed, or advanced discrete transform. Data holds including the zero order, first order, second order, and slewer can be inserted into the transformation. The second program, TXCONV, implements the conversion of a high-rate discrete transform to a low-rate discrete transform. The general input/output structure of these two computer programs is shown in Fig. 1. In DISCRET, the input is an s-plane transfer function and the

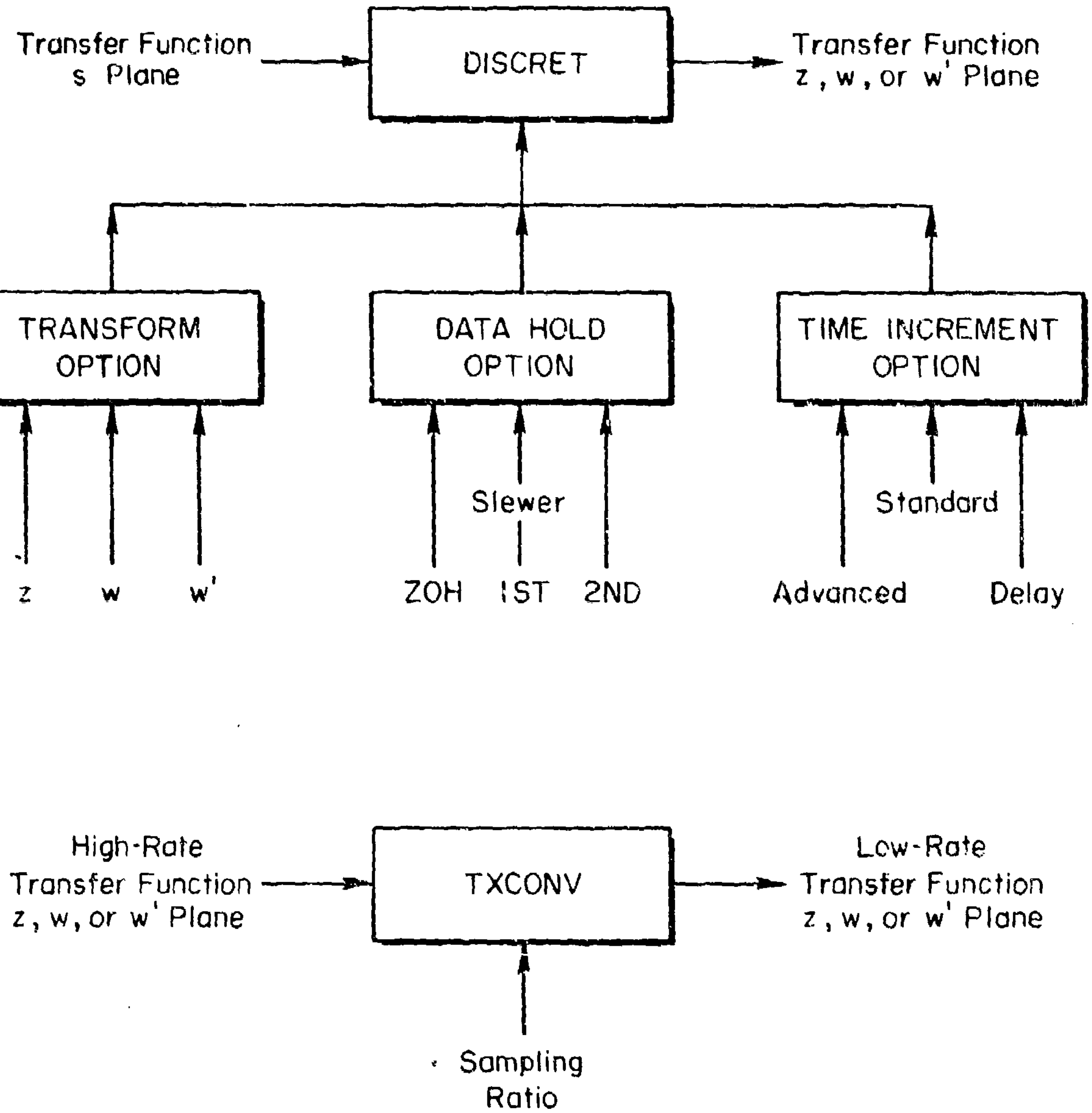


Figure 1. General Structure of DISCRET and TXCONV Computer Programs

output is a selected discrete transfer function. For TXCONV, the input is a high-rate discrete transfer function and the output a low-rate discrete transfer function in the z-, w-, w′-plane.

In Section II, a review of basic sampled-data theory is presented. This section provides the necessary background information for succeeding sections. The mathematical development is based on the assumption that the sampling process can be described as the amplitude modulation of an impulse train by the input signal. This assumption greatly simplifies sampled-data theory and is valid for most practical engineering applications.

The transform conversion expressions mechanized in the TXCONV computer program are developed in Section III. These expressions allow a high-rate transform in the z-, w-, or w′-plane to be converted to a low-rate discrete transform. The fundamental definition of the z-transform and the discrete inversion integral developed in Section II form the basis for the analytical development. The computer mechanization of the transform conversion is based on the practical calculation of the residues of a complex integral. The residues for this integral are calculated in an unconventional manner using a limiting process via L'Hôpital's rule. This method simplifies the mechanization scheme and leads to a closed-form solution.

Sections IV and V provide a detailed description of the DISCRET and TXCONV computer programs, respectively. This includes the available program options, theoretical basis for the mechanization algorithms, general and detailed program structure, required input data, and typical program output. Source listings for these two computer programs are contained in Volume III.

SECTION II

REVIEW OF FUNDAMENTAL SAMPLED-DATA THEORY

A. INTRODUCTION

A quick review of the fundamental principles of sampled-data theory is presented in this section (Refs. 1-10). This background information will be used in succeeding sections to develop the analytical expressions mechanized in the DISCRET and TXCONV computer programs. The basic theory for sampled-data or discrete systems was developed over 20 years ago and remains intact today. Practical sampled-data theory is based on the assumption that the actual sampling operation can be modeled as the amplitude modulation of an impulse train. This central concept greatly simplifies the analysis and synthesis of sampled-data systems. Fortunately, this view of the sampling process is valid for most practical systems and use of this theory is normally considered exact.

Sampled-data systems generally contain both continuous and discrete elements. The z-transform provides a unified analysis and synthesis technique for these hybrid systems. For a sampled continuous element, the z-transform can be considered as the Laplace transform of an impulse sequence (impulse train) where the area or strength of the individual impulses equal the value of the continuous time function at each discrete sampling instant. An alternate viewpoint is to consider the exponent in the z^{-n} delay operator as an ordering variable for a number sequence (or a sequence of discrete signal values) where the coefficient for the z^{-n} terms equals the value of the number sequence (or the discrete signal) at the nth discrete time instance. This viewpoint allows the time domain difference (or recursion) equation that describes the number sequence (or sequence of discrete signal values) to be modeled in the z-plane. In practice, the continuous functions in a hybrid system are first expressed in the s-domain and then transformed to the z-plane using standard techniques such as partial fraction expansion coupled with table lookup or by employing the inversion integral and contour

integration. (The partial fraction expansion table lookup approach is mechanized in the DISCRET computer program.) On the other hand, a discrete function (e.g., digital controller) may be first modeled with a recursion equation and then directly converted to the z-plane by substituting the z^{-n} delay operator for each discrete term in the recursion equation. Naturally, during the design phase, it is the discrete controller expressed in the z-plane that is first obtained and then converted to a recursion equation using the z^{-n} delay operator and subsequently implemented on a digital computer.

In analysis and design, no distinction is normally necessary between the z-transform function derived from a sampled continuous element and the z-transform function that models a completely discrete element. Once discretized, all elements of a hybrid system can be treated using common analysis and design techniques and tools. However, consideration must be given to the fact that discretizing a continuous function in the z-domain only accounts for the continuous variables at the sampling instance. In general, the inter-sample response is not modeled with the standard z-transform. It is necessary to investigate the inter-sample response using such techniques as the continuous frequency response and T/N methods in Ref. 1 or the advanced (or delayed) z-transform. Nevertheless, the ability to model continuous and discrete elements in a common domain is one of the most fundamentally useful properties of the z-transform (and the w- or w′-transform).

B. FUNDAMENTAL SAMPLED-DATA RELATIONSHIPS

The fundamental relationships for the Laplace transform of a sampled signal are:

$$C^T(s) = \sum_{k=0}^{\infty} c(kT)\, e^{-kTs} \qquad (1)$$

$$C^T(s) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} C(p) \frac{1}{1 - e^{-T(s-p)}} \, dp \tag{2}$$

$$C^T(s) = \frac{1}{T} \sum_{k=-\infty}^{\infty} C\left(s - \frac{j2\pi k}{T}\right) \tag{3}$$

The superscript T designates the time interval between each sampling operation. These expressions are equivalent and each has found varying degrees of utility in sampled-data or discrete system theory. All assume that the sampling process can be visualized as the amplitude modulation of an impulse train $\delta_T(t)$ by the input signal (Fig. 2). The impulse train (Eq. 4) represents a series of impulses of unit strength or area equally spaced in time and extending from zero to plus infinity.

$$\delta_T(t) = \delta(t) + \delta(t - T) + \delta(t - 2T) + \cdots = \sum_{k=0}^{\infty} \delta(t - kT) \tag{4}$$

The Laplace transform of $\delta_T(t)$ is given in closed form as

$$\mathcal{L}[\delta_T(t)] = 1 + e^{-sT} + e^{-2sT} + \cdots = \sum_{k=0}^{\infty} e^{-ksT}$$

$$= \frac{1}{1 - e^{-sT}} \quad , \quad |e^{-sT}| < 1 \tag{5}$$

Equation 1 is the standard definition of the z-transform with the simple change of variable

$$z = e^{sT} \tag{6}$$

where T is the sampling interval. Substituting Eq. 6 into Eq. 1 converts $C^T(s)$, a nonalgebraic function in s, to a rational function in z.

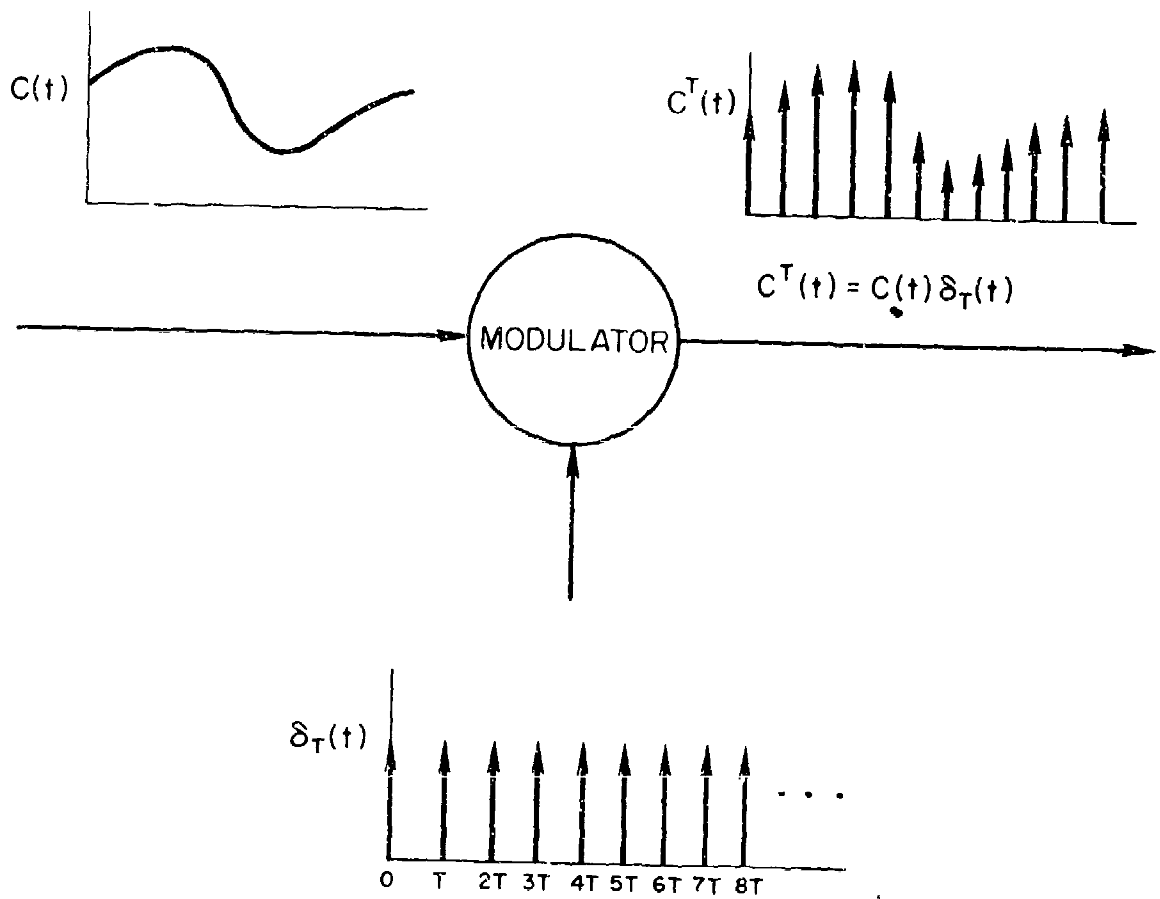


Figure 2. Amplitude Modulation of Impulse Train by Continuous Signal

This change of variable allows many of the well-defined analysis and synthesis techniques developed for continuous systems to be applied more readily to sampled-data systems.

The relationship in Eq. 2 is obtained by exercising the method of complex convolution. The dual relationships which are of fundamental importance are stated below:

- The Laplace transform of the convolution of two time functions is equal to the product of their individual transforms.
- The Laplace transform of the multiplication of two time functions is equal to the convolution of their transforms in the complex domain.

An analytical definition of the latter relationship is expressed as

$$G(s) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} G_1(p)\ G_2(s-p)\ dp \tag{7}$$

where

$$g(t) = g_1(t)\ g_2(t) \quad , \quad \sigma_{a1} < c < \mathrm{Re}(s - \sigma_{a2}) \tag{8}$$

For convergence, the real part of s must be large enough so that all the poles of $G_2(s - p)$ in the p-plane lie to the right of the poles of $G_1(p)$. The abscissa of absolute convergence of $G_1(p)$ and $G_2(s - p)$ are, respectively, σ_{a1} and σ_{a2}. Applying Eq. 7 to the sampling process represented by the multiplication of an impulse train by a continuous time function [i.e., $C^T(t) = c(t)\delta_T(t)$], and recalling that the Laplace transform of $\delta_T(t)$ is given by Eq. 5, results in Eq. 2.

There are many ways of deriving Eq. 3. Assuming C(s) has at least two more poles than zeros, or the initial value of c(t) is zero [i.e., C(s) has a continuous impulse response], then the open interval of integration in Eq. 2 may be closed through an infinite semicircle in the right half plane as shown in Fig. 3. The integral along the infinite semicircle vanishes as a consequence of the assumption that the degree of the denominator of C(s) is at least two higher than the numerator

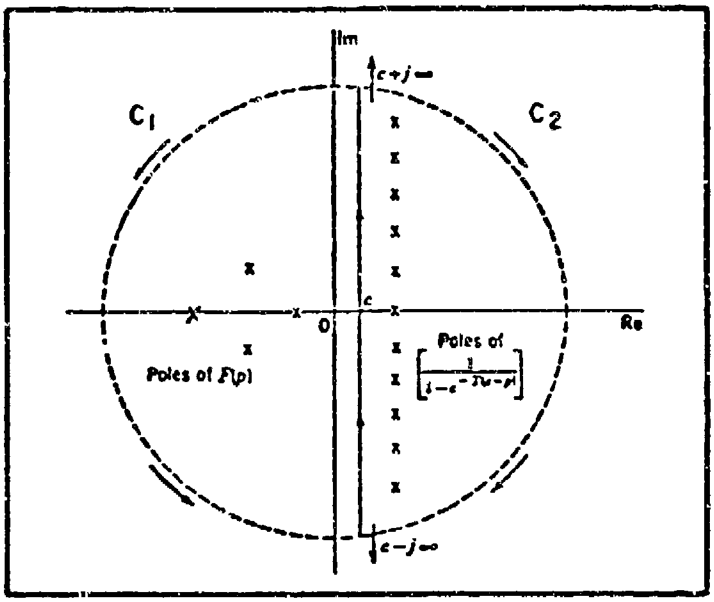


Figure 3. The Complex p-Plane

(Ref. 5). The closed contour integration then reduces to the original line integral in Eq. 2. Cauchy's integral formula then allows the evaluation of Eq. 2 within the closed contour C_2 as an infinite summation of residues which include all the poles of

$$\frac{1}{1 - e^{-T(s-p)}} = 0 \tag{9}$$

or

$$p = s - \frac{j2\pi k}{T} \quad , \qquad k = 0, \pm 1, \pm 2, \ldots \tag{10}$$

Equation 2 then reduces to

$$C^T(s) = - \sum_{k=-\infty}^{\infty} \left. \frac{C(p)}{\frac{d}{dp}\left[1 - e^{-T(s-p)}\right]} \right|_{p=s-(j2\pi k/T)} \tag{11}$$

The negative sign for the summation is a result of the clockwise contour C_2. Evaluating the derivative in the denominator results in

$$\frac{d}{dp}\left[1 - e^{-T(s-p)}\right]\Bigg|_{p=s-(j2\pi k/T)} = -Te^{j2\pi k} = -T \tag{12}$$

Equation 11 then reduces to

$$C^T(s) = \frac{1}{T}\sum_{k=-\infty}^{\infty} C\left(s - \frac{j2\pi k}{T}\right) \tag{13}$$

If $C(s)$ has a denominator one degree higher than its numerator or $c(0^+) \neq 0$, Eq. 13 should be modified to include an additional initial condition term (Ref. 5).

$$C^T(s) = \frac{1}{T}\sum_{k=-\infty}^{\infty} C\left(s - \frac{j2\pi k}{T}\right) + \frac{1}{2}\,c(0^+) \tag{14}$$

Restricting $C(s)$ to be of order $1/s^m$, where $m > 2$ in Eq. 13 and $m > 1$ in Eq. 14, insures that the infinite summation will be absolutely convergent and independent of the order of summation. However, the restriction on Eq. 13 can be relaxed to $m > 1$ if the sum is evaluated by taking pairs of terms corresponding to equal positive and negative values of the index k. Under this condition, the sum in Eq. 13 will then be absolutely convergent (Ref. 11).

An alternate expression for $C^T(s)$ can be obtained from Eq. 2 by closing the contour to the left and evaluating the finite residues of $C(p)$. This contour avoids the problems of an infinite summation. For this case, $C(s)$ is required only to have a denominator one degree higher than its numerator. Under these conditions, Eq. 2 reduces to the following finite summation of residues corresponding to the poles of $C(p)$ in the p-plane.

$$C^T(s) = \sum_k \text{residues} \left. \frac{C(p)}{1 - e^{-T(s-p)}} \right|_{p=\text{Poles of } C(p)} \qquad (15)$$

C. z-TRANSFORM AND THE INVERSION INTEGRAL

The analytical derivation of a general conversion equation which allows a low-rate discrete transform to be calculated from a high-rate discrete transform (TXCONV computer program), relies on the fundamental definition of the z-transform and application of the discrete inversion integral. These two relationships are derived in some detail in this subsection.

The definition of the z-transform stems from the infinite summation

$$C^T(t) = \sum_{k=0}^{\infty} c(kT)\ \delta(t - kT) \qquad , \qquad k = 0, 1, 2, \ldots \qquad (16)$$

where $C^T(t)$, the sampled signal, is represented by the area or strength of impulses equal to the magnitude of $c(t)$ at the sampling instants $t = kT$. Viewing the sampling process as the amplitude modulation of an impulse train $\delta_T(t)$ by the signal $c(t)$ at the sampling instants forms the mathematical basis for practical sampled-data system analysis and synthesis. Such a viewpoint is justified if the actual time during which the sampler is closed is short compared to the time constants in the system under investigation. It is shown in Ref. 5 that for a system with a single time constant $\tau = 1/a$, the error using impulse modulation is less than 5 percent for a sampler pulse width h which is less than or equal to one-tenth of the time constant (i.e., $h/\tau \leq 1/10$). It is significant to note that whether $c(t)$ is sampled physically or fictitiously, or already exists in pulsed form, $C^T(t)$ is still representative of an equivalent linearized continuous signal $c(t)$ at the sampling instants $t = kT$. This point will be elaborated on in the next subsection. Taking the Laplace transform of Eq. 16 produces

$$C^T(s) = c(0) + c(T)e^{-sT} + c(2T)e^{-2sT} + \cdots = \sum_{k=0}^{\infty} c(kT)e^{-ksT} \qquad (17)$$

In general, if the Laplace transform of c(t) is a rational algebraic function, a closed form can be found for the infinite series representation of $C^T(s)$. The final simple change in variable $z = e^{sT}$ results in the one-sided z-transform

$$C^T(z) = \sum_{k=0}^{\infty} c(kT)z^{-k} \qquad (18)$$

For the two-sided z-transform, the lower summation limit becomes minus infinity and c(t) is defined for negative time.

The inversion integral is a closed form technique for finding the inverse z-transform (Eq. 19).

$$c(kT) = \frac{1}{2\pi j} \oint_{C_1} z^{k-1}C^T(z)\, dz \qquad (19)$$

Equation 19 is based on the Laurent series expansion of $F(z) = z^{k-1}C^T(z)$ about $z = 0$. Expanding Eq. 18, the fundamental definition of the z-transform, produces

$$C^T(z) = c(0) + c(T)z^{-1} + c(2T)z^{-2} + \cdots + c(kT)z^{-k} + \cdots \qquad (20)$$

If we now multiply Eq. 20 by z^{k-1},

$$F^T(z) = z^{k-1}C^T(z) = c(0)z^{k-1} + c(T)z^{k-2} + \cdots + c(kT)z^{-1} + \cdots \qquad (21)$$

The desired output c(kT) in the Laurent series expansion is defined as the residue of the function $F^T(z)$. This result may be generalized through application of the Cauchy theorem which states that if the integral $F^T(z)$ is defined by

$$F^T(z) = \frac{1}{2\pi j} \oint_{C_1} z^k \, dz \qquad (22)$$

and the integral is taken around a closed contour C_1 which encloses the origin of the z-plane, then $F^T(z)$ is given by

$$\begin{aligned} F^T(z) &= 0 \quad , \quad k < -1 \\ F^T(z) &= 1 \quad , \quad k = -1 \qquad (23) \\ F^T(z) &= 0 \quad , \quad k > -1 \end{aligned}$$

where the $k = -1$ case is recognized as the residue of $F^T(z)$. Applying this theorem term by term to Eq. 21 results in the discrete inversion integral, Eq. 19. The desired discrete time inversion for $c(kT)$ then reduces to the practical evaluation of the residues of the poles associated with $[z^{k-1}C^T(z)]$ expressed in closed form as

$$c(kT) = \sum \text{residues of } \left[z^{k-1}C^T(z)\right] \text{ at poles of } z^{k-1}C^T(z) \qquad (24)$$

SECTION III

DEVELOPMENT OF TRANSFORM CONVERSION EXPRESSION

A. INTRODUCTION

The TXCONV computer program is mechanized to calculate a low-rate discrete transform from a given high-rate discrete transform. This section analytically derives the transform conversion expressions used in TXCONV. Detailed derivations are given in the z-, w-, and w′-planes. The mechanization scheme in TXCONV is based on the practical calculation of the residues associated with a unique form of the inversion integral derived herein. The methodology presented in this section is exact for integer ratios of high-to-low sampling rate and is based on an equivalent linearized continuous response for non-integer ratios.

B. TRANSFORM CONVERSION IN z-DOMAIN

The objective of the following mathematical development is to derive a closed-form expression for the low-rate transform $C^T(z)$ as a function of the high-rate transform $C^{T/N}(z_p)$ represented by

$$C^T(z) = \left[C^{T/N}(z_p)\right]^T \qquad (25)$$

This transformation inherently arises when the output of a system is sampled at a lower rate than the input (Fig. 4). The superscript in Eq. 25 designates the sampling interval in the z or z_p transform. That is,

$$C^T(z) \text{ --- } z = e^{sT} \qquad (26)$$

$$C^{T/N}(z_p) \text{ --- } z_p = e^{sT/N} \qquad (27)$$

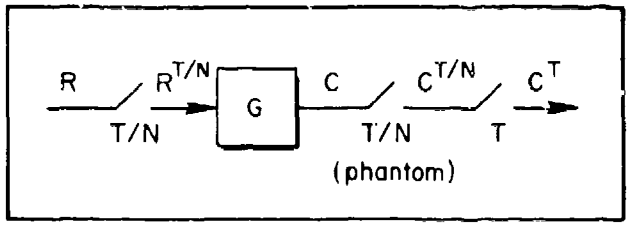


Figure 4. Fast-Input/Slow-Output Sampling with Phantom T/N Output Sampler

The z_p-transform of the sampled signal $C^{T/N}(t)$ is first calculated with respect to a sampling interval of T/N producing the high-rate transform $C^{T/N}(z_p)$. Then, the z-transform of this high-rate transform is taken with respect to a T sampling interval producing the low-rate transform $C^T(z)$. This constitutes the general form of the transform conversion addressed in this subsection. To simplify the notation, the z and z_p designation will be suppressed and C^T and $C^{T/N}$ will be used to designate the z and z_p transforms.

We will assume that the sampling ratio N in Eq. 25 can be any integer or non-integer rational value. However, only integer values of N are allowed in most practical sampled-data systems. More will be said about this in the next subsections. For the present, we proceed with the derivation for rational values of the sampling ratio and subsequently treat integer values as a special case of the more general non-integer case.

The respective sampled signals in Eq. 25 are defined in the z-domain using Eq. 18.

$$C^{T/N} = \sum_{k=0}^{\infty} c(kT/N) z_p^{-k} \quad , \quad z_p = e^{sT/N} \tag{28}$$

$$C^{T} = \sum_{k=0}^{\infty} c(kT) z^{-k} \quad , \quad z = e^{sT} \tag{29}$$

Transforming Eq. 28 back into the time domain using the inversion integral (Eq. 19) results in

$$c(kT/N) = \frac{1}{2\pi j} \oint_{C_1} C^{T/N} z_p^{k-1} dz_p \tag{30}$$

It is important to recognize that although Eq. 30 provides information only at discrete instances of time separated by T/N seconds, a linearized continuous time function c(t) can be obtained from the solution of Eq. 30 by replacing kT/N with t. This linearized system response agrees with the sampled response at the sampling instants $t = kT/N$. Moreover, this linearized response also exactly characterizes the low-rate sampled response c(kT) with t replaced by kT for integer values of N. It approximates the low-rate sampled response c(kT) for non-integer values of N by assuming that c(kT/N) is the high-rate sampled response of a continuous system c(t). For example, if C(s) contains only simple poles at $(a_1, a_2, a_3, \ldots)$, the general closed-form discrete time solution from Eq. 30 is represented by

$$c(kT/N) = A_1 e^{-a_1 kT/N} + A_2 e^{-a_2 kT/N} + A_3 e^{-a_3 kT/N} + \cdots \tag{31}$$

where $(A_1, A_2, A_3, \ldots)$ represent the residues of the simple poles. If kT/N is replaced by t in Eq. 31, the linearized continuous response is obtained (Eq. 32) which exactly matches the sampled response c(kT/N) at $t = kT/N$.

$$c(t) = A_1 e^{-a_1 t} + A_2 e^{-a_2 t} + A_3 e^{-a_3 t} + \cdots \tag{32}$$

It is readily apparent that sampled responses at other than the T/N interval can be obtained from Eq. 30 via a change in the ordering index k. For the T sampling interval, substituting $k = kN$ in Eq. 30 produces

$$c(kT) = \frac{1}{2\pi j} \oint_{C_1} C^{T/N} z_p^{kN-1}\, dz_p \tag{33}$$

Substituting Eq. 33 into Eq. 29 results in Eq. 34.

$$C^T = \sum_{k=0}^{\infty} \left[\frac{1}{2\pi j} \oint_{C_1} C^{T/N} z_p^{kN-1}\, dz_p \right] z^{-k} \tag{34}$$

Interchanging the summation and integration in Eq. 34 produces Eq. 35.

$$C^T = \frac{1}{2\pi j} \oint_{C_1} C^{T/N} \left[\sum_{k=0}^{\infty} \left(z_p^N z^{-1} \right)^k \right] \frac{dz_p}{z_p} \tag{35}$$

The infinite summation in Eq. 35 is recognized as a geometric progression in $\left(z_p^N z^{-1}\right)$ which can be placed in closed form as indicated in Eq. 36.

$$C^T = \frac{1}{2\pi j} \oint_{C_1} C^{T/N} \frac{1}{1 - z_p^N z^{-1}} \frac{dz_p}{z_p} \tag{36}$$

Equation 36 is the final result which can be used to calculate any general low-rate z-transform from a given high-rate z_p-transform. To evaluate Eq. 36, the integration contour C_1 in the z_p-plane can be selected to include all the poles of $C^{T/N}/z_p$ and exclude the N poles of $\left(z_p^N z^{-1}\right)$. Alternatively, the contour C_1 can include only the N poles of $\left(z_p^N z^{-1}\right)$ and exclude the poles associated with $C^{T/N}/z_p$. For either approach, the problem reduces to calculating the residues of the enclosed poles. The computationally more convenient method is to evaluate the residues of $C^{T/N}/z_p$. Equation 36 then reduces to the finite summation given in Eq. 37.

$$C^T = \sum_n \text{residues } \frac{C^{T/N}}{z_p} \left. \frac{z}{z - z_p^N} \right|_{z_p = \text{Poles of } C^{T/N}/z_p} \qquad (37)$$

where

$$C^T = C^T(z) \qquad \text{Low-rate transform, } z = e^{sT}$$

$$C^{T/N} = C^{T/N}(z_p) \qquad \text{High-rate transform, } z_p = e^{sT/N}$$

C. TRANSFORM CONVERSION IN w- AND w′-PLANE

The bilinear transformation from the z-plane to the w- and w′-planes are defined by:

$$z = \frac{1 + w}{1 - w} \qquad w = \frac{z - 1}{z + 1} \qquad (38)$$

$$z = \frac{2/T + w'}{2/T - w'} \qquad w' = \frac{2}{T} \frac{z - 1}{z + 1} \qquad (39)$$

Since w' is related to w by a scale factor $2/T$, the bilinear transformation can be expressed as

$$z = \frac{A + w}{A - w} \qquad (40)$$

where w represents either w or w' and $A = 1$ for w and $A = 2/T$ for w'. Substituting Eq. 40 into Eqs. 28 and 29 produces

$$C^{T/N} = \sum_{k=0}^{\infty} c(kT/N) \left[\frac{A_p + w_p}{A_p - w_p} \right]^{-k} , \quad w_p = A_p \frac{z_p - 1}{z_p + 1} \qquad (41)$$

$$C^T = \sum_{k=0}^{\infty} c(kT) \left[\frac{A + w}{A - w} \right]^{-k} , \quad w = A \frac{z - 1}{z + 1} \qquad (42)$$

where A is associated with the low-rate transform and A_p with the high-rate transform. Transforming Eq. 30 into the w- or w′-plane involves a change in the integrtion variable given by

$$dz_p = \frac{2A_p}{(A_p - w_p)^2} \, dw_p \tag{43}$$

Substituting Eqs. 40 and 43 into Eq. 30 and simplifying produces

$$c(kT/N) = \frac{1}{2\pi j} \oint_{C_1} C^{T/N} \left[\frac{A_p + w_p}{A_p - w_p}\right]^k \frac{2A_p}{A_p - w_p} \, \frac{dw_p}{A_p + w_p} \tag{44}$$

Changing the sampling index to k = kN in Eq. 44, substituting Eq. 44 into Eq. 42, and interchanging the summation and integration results in

$$C^T = \frac{1}{2\pi j} \oint_{C_1} C^{T/N} \left\{ \sum_{k=0}^{\infty} \left[\left(\frac{A_p + w_p}{A_p - w_p}\right)^N \left(\frac{A + w}{A - w}\right)^{-1} \right]^k \right\} \frac{2A_p}{A_p - w_p} \, \frac{dw_p}{A_p + w_p} \tag{45}$$

Placing the infinite summation in closed form reduces Eq. 45 to

$$C^T = \frac{1}{2\pi j} \oint_{C_1} C^{T/N} \frac{1}{1 - X} \, \frac{2A_p}{A_p - w_p} \, \frac{dw_p}{A_p + w_p} \tag{46}$$

where

$$X = \left[\left(\frac{A_p + w_p}{A_p - w_p}\right)^N \left(\frac{A + w}{A - w}\right)^{-1} \right] \tag{47}$$

and

$$|X| < 1 \tag{48}$$

As was done in Eq. 36, a closed contour C_1 is taken which includes all the poles of the integrand in Eq. 46 except the poles associated with the infinite summation term $1/1 - X$. The free $1/A_p - w_p$ term must also be excluded since it is in the same contour region as the $1/1 - X$ term. It is required that the infinite sum in Eq. 45 be absolutely convergent (i.e., satisfy Eq. 48) over the region of integration. For then, the integral remains finite and the summation of integrals in Eq. 45 equals the integral of the summation. This condition assures a closed form solution for Eq. 46. The integral then reduces to a summation of residues given by:

$$C^T = \sum_n \text{residues} \left. \frac{C^{T/N}}{A_p + w_p} \frac{1}{1 - X} \frac{2A_p}{A_p - w_p} \right|_{w_p = \text{Poles of } C^{T/N}/(A_p + w_p)} \tag{49}$$

where

w_p = High-rate transform variable

w = Low-rate transform variable

$w = w'$, $w_p = w_p'$ for $A_p = 2N/T$ and $A = 2/T$

$w = w$, $w_p = w_p$ for $A_p = 1$ and $A = 1$

D. GENERAL MULTI-RATE CONFIGURATIONS

In the preceding derivation leading to Eq. 36 (and Eq. 49) it was shown that the high to low-rate sampling ratio N can be any rational value. There are three general multi-rate cases of particular interest in sampled-data systems:

- If N is an integer, the result obtained from Eqs. 36 or 49 are exact for all multi-rate configurations. This is easily seen since the exact discrete low-rate information c(kT) can be selectively extracted from the original high-rate discrete signal c(kT/N). Therefore, no use is made of the linearized continuous response function.

- For non-integer values of N, the results from Eqs. 36 or 49 are exact if the original high-rate discrete response $c(kT/N)$ is the sampled response from a completely continuous linear function $c(t)$ or $C(s)$.

- If N is a non-integer and $c(kT/N)$ is the response from a high-rate discrete function (e.g., digital computer algorithm), the results from Eqs. 36 or 49 are at best approximate since the low-rate response $c(kT)$ is based on a linearized continuous model of the original discrete function.

The most practical multi-rate configurations are those which can be analyzed with integer values of N. The high-rate transform [illegible]$^{/N}$ can originate from a completely continuous function, a completely discrete function, or any combination of continuous and discrete functions, and Eqs. 36 or 49 provide the exact low-rate discrete transform for integer ratios of sampling rate.

Consider the fundamental fast-input/slow-output multi-rate sampling configuration in Fig. 5 where M represents the transfer function of a data hold and G a continuous system in the s-plane. In general, the output sampling interval T_2 is greater than the input sampling interval T_1; however, a direct integer ratio between output-to-input sampling is not necessarily implied. This simple multi-rate configuration represents a complex (but practical) situation which can be analyzed using a common sampling period T such that

$$\frac{T}{M} = T_1 \qquad \frac{T}{N} = T_2 \qquad M,N = \text{integer} \tag{50}$$

For example, if the input is sampled at 20 cps and the output at 13.33 cps; $T_1 = 0.050$, $T_2 = 0.075$, and a choice of $T = 0.150$ produces

$$\frac{T}{3} = T_1 \qquad \frac{T}{2} = T_2 \qquad T = 0.150 \tag{51}$$

The multi-rate configuration in Fig. 5 then reduces to the general form in Fig. 6. The output equation for Fig. 6 is given by

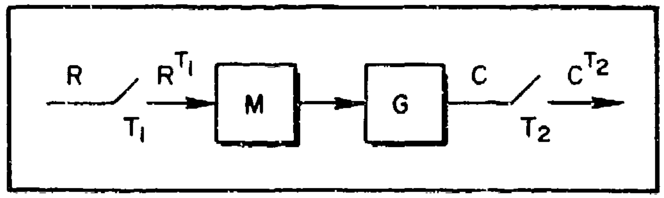


Figure 5. Fundamental Fast-Input/Slow-Output Multi-Rate Sampling, $T_2 > T_1$

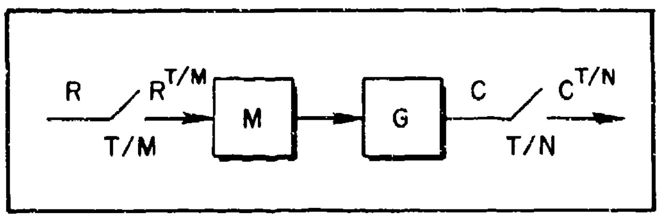


Figure 6. Fast-Input/Slow-Output Sampling with Common Sampling Period T

$$C^{T/N} = [GMR^{T/M}]^{T/N} \tag{52}$$

Computationally, Eq. 52 is more involved than a general slow-input/fast-output system, since the T/N operator does not "operate through" the non-integer ratio of inner-to-outer sampling. That is, the T/N sampling interval is greater than the inner sampling interval T/M. Moreover, T/N is not necessarily an integer multiple of T/M, which further complicates the analysis. Fortunately, we are free to add a mathematical or phantom sampler to the output (Fig. 7) which operates at an integer multiple of the output sampling rate or at a submultiple of the output sampling interval T/N. This mathematical convenience is valid since the actual output sampler T/N simply rejects all the unwanted samples from the phantom sampler T_*. This simplification overcomes the above complications and facilitates a solution to Eq. 52 using the high-rate to low-

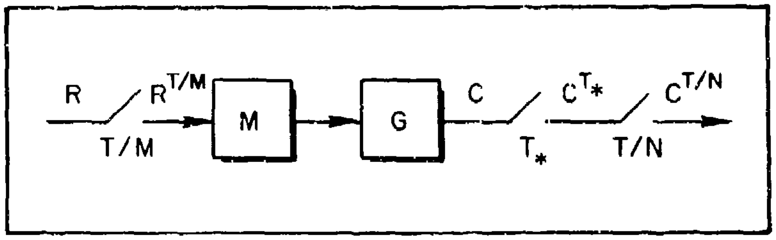


Figure 7. Fast-Input/Slow-Output Sampling with Phantom Sampler T_*

rate transform expression in Eqs. 36 or 49. With the additional sampler T_*, the output equation from Fig. 7 becomes

$$C^{T/N} = \left[(GM)^{T_*} R^{T/M} \right]^{T/N} \tag{53}$$

The problem is now reduced to finding the individual transforms $(GM)^{T_*}$ and $R^{T/M}$ (and their product) using a common definition for the transform variable z, w, or w′. For the more general case of where M/N is not an integer, a value for T_* must be selected such that T_* is both smaller than and an integer submultiple of T/M and T/N. An obvious choice is $T_* = T/MN$; however, the largest compatible value for T_* is composed of the prime factors of the integer product MN. The smaller T_* is, the higher the order of the numerator and denominator polynomials in the $R^{T/M}$ term. This increase in order is the result of substituting the higher rate transform variable (e.g., $z_* = e^{sT_*}$) associated with the $(GM)^{T_*}$ term into the lower rate $R^{T/M}$ term ($z_m = e^{sT/M}$) to form an overall high-rate discrete transform product. For example, if $z_* = e^{sT/6}$ and $z_m = e^{sT/3}$, then the z_m transform variable can be defined by $z_m = z_*^2$ and the $(GM)^{T/6}R^{T/3}$ product can be formed using the common transform variable z_*. Therefore, a T_* composed of the prime factors of MN reduces to a minimum the resulting order of the $R^{T/M}$ term. This in turn may reduce the computations required to calculate the low-rate T/N transform in Eq. 53 (using Eqs. 36 or 49) since the number of residues in the $(GM)^{T_*}R^{T/M}$ product will be at a minimum. Using the prime notation, the general sampling structure in Eq. 53 becomes

$$C^{T/N} = \left[(GM)^{T/MN*} R^{T/M}\right]^{T/N} \tag{54}$$

Inserting the previous numerical sampling values into Eq. 54 will help summarize and clarify the general results.

$$C^{T/2} = \left[(GM)^{T/6} R^{T/3}\right]^{T/2} \tag{55}$$

The selection of $T_* = T/6$ allows the formation of the inner transform product $(GM)^{T/6}R^{T/3}$ and at the same time provides an integer ratio of outer-to-inner sampling periods. Equation 55 is now in a form that can be easily solved by Eqs. 36 or 49.

A second fundamentally important system configuration is shown in Fig. 8. Here, $G^{T/N}$ represents a discrete model of the digital computations in a computer (e.g., digital control laws). M_N is a data hold device that models the holding of information in a storage register between sampling intervals. The actual computational time in the computer is assumed negligible in this case and is not considered. However, computational delays can be easily handled with appropriate delay factors and the advanced z-, w-, or w′-transforms. Simple algebraic signal flow tracing produces Eq. 56, the output equation for Fig. 8.

$$C^{T} = \left[M_n G^{T/N} R^{T/N}\right]^{T} \tag{56}$$

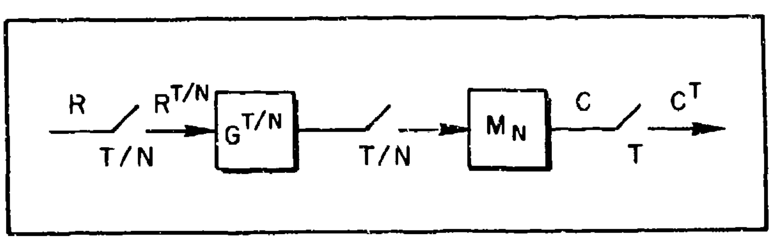


Figure 8. Fast-Input/Slow-Output Sampling with Discrete System Component

The same situation persists here as in Eq. 52 and we are unable to operate through with T. Adding a phantom sampler to the output allows us to write

$$C^T = \left[M^{T/N} G^{T/N} R^{T/N}\right]^T \tag{57}$$

The C^T transform in the z-plane can now be easily obtained from Eq. 36 by evaluating the residues of the inner transform product with the transform variable defined as $z = e^{sT/N}$.

SECTION IV

DESCRIPTION OF DISCRET COMPUTER PROGRAM

A. INTRODUCTION

DISCRET is a versatile and relatively accurate digital computer program that transforms a continuous s-plane transfer function into a valid discrete transfer function in the s-, w-, or w′-plane. The program calculates the discrete transformation for a system of the general form depicted in Fig. 9.

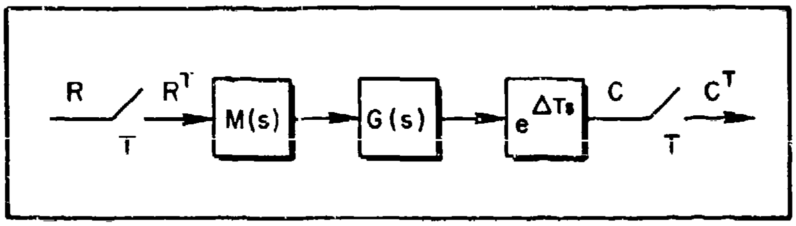


Figure 9. General Form of Sampled System

M(s) in Fig. 9 represents the s-plane transfer function model of a data hold, G(s) the s-plane representation of a continuous system, and $e^{\Delta Ts}$ the time factor in the advanced or delayed discrete transform. For the standard z-, w-, or w′-transform, $\Delta T = 0$. The output equation for Fig. 9 is

$$C^T = \left[e^{\Delta Ts}\, G(s)M(s)\right]^T R^T \qquad (58)$$

The superscript T denotes the sampling period in the z-, w-, or w′-transform (i.e., $z = e^{sT}$). The DISCRET program calculates the general transform given by

$$\left[e^{\Delta Ts}\, G(s)M(s)\right]^{T} \tag{59}$$

This general transform is present in most open-loop and closed-loop sampled-data or discrete systems.

DISCRET is written in FORTRAN for the Control Data Corporation (CDC) CYBER 175 series computer. It can handle pole multiplicity up to three and system order up to 50th. Double precision arithmetic is used throughout the program.

B. PROGRAM OPTIONS

DISCRET calculates the practical discrete transformations required to analyze and design realistic sampled-data systems. The program can execute any combination of the following options:

- Transform Options
  - z-transform
  - w-transform
  - w′transform
- Data Hold Options
  - None
  - Zero order hold
  - First order hold
  - Second order hold
  - Slewer
- Time Increment Option
  - Standard transform
  - Delayed transform
  - Advanced transform

C. MATHEMATICAL DESCRIPTION OF PROGRAM

The computer code in DISCRET is based on the computer program in Ref. 12. Extensions and modifications have been made to accommodate additional options. The analytical basis for the computer algorithms is discussed in Ref. 12. However, to provide a complete description of the program, some of the details are repeated.

Consider the partial fraction expansion of the general expression in Eq. 59.

$$\left[e^{\Delta Ts}G(s)M(s)\right]^T = F^T(z)\left\{e^{\Delta Ts}\left[\frac{K_1}{(s+a_1)} + \frac{K_{21}}{(s+a_2)} + \frac{K_{22}}{(s+a_2)^2} + \frac{K_{31}}{(s+a_3)} + \frac{K_{32}}{(s+a_3)^2} + \frac{K_{33}}{(s+a_3)^3} + \cdots\right]\right\}^T \tag{60}$$

$F^T(z)$ in Eq. 60 is a z-plane term determined by the data hold selected. For example, for a zero order hold (ZOH), Eq. 59 becomes

$$\left[e^{\Delta Ts}G(s)M(s)\right]^T = \left[\frac{1-e^{-sT}}{s}\, e^{\Delta Ts}G(s)\right]^T = \frac{z-1}{z}\left[e^{\Delta Ts}\,\frac{G(s)}{s}\right]^T \tag{61}$$

where the ZOH introduces a pole at $s = 0$ and a z-plane factor $z - 1/z$. For other data holds, a similar situation exists. This can be readily verified by applying the data hold transfer functions in Table 1 to Eq. 59.

TABLE 1. DATA HOLDS IMPLEMENTED IN DISCRET

DATA HOLD	TRANSFER FUNCTION
Zero-Order Hold	$M_0 = \dfrac{1 - e^{-sT}}{s}$
First-Order Hold	$M_1 = M_0^2\left(s + \dfrac{1}{T}\right)$
Second-Order Hold	$M_2 = M_0^2\left(s^2 + \dfrac{3}{2T}\, s + \dfrac{1}{T^2}\right)$
Slewer Hold	$M_{slew} = \dfrac{M_0^2}{T}$

In DISCRET, the individual partial fraction expansion terms in Eq. 60 are first transformed into the w-plane using the following advanced w-transforms:

$$\left[\frac{e^{\Delta Ts}}{s + a}\right]^T = \frac{e^{-a\Delta T}}{1 + e^{-aT}}\left\{\frac{1 + w}{w + \left(1 - e^{-aT}/1 + e^{-aT}\right)}\right\} \tag{62}$$

$$\left|\frac{e^{\Delta Ts}}{(s + a)^2}\right|^T = \frac{\Delta Te^{-a\Delta T}}{1 + e^{-aT}}\left\{\frac{1 + w}{w + \left(1 - e^{-aT}/1 + e^{-aT}\right)}\right\}$$

$$+ \frac{Te^{-aT}e^{-a\Delta T}}{\left(1 + e^{-aT}\right)^2}\left\{\frac{(1 + w)(1 - w)}{\left[w + \left(1 - e^{-aT}/1 + e^{-aT}\right)\right]^2}\right\} \tag{63}$$

$$\left[\frac{e^{\Delta Ts}}{(s+a)^3}\right]^T = \frac{(\Delta T)^2 e^{-a\Delta T}}{2\left(1+e^{-aT}\right)}\left\{\frac{1+w}{w+\left(1-e^{-aT}/1+e^{-aT}\right)}\right\}$$

$$+\frac{T^2(1+2\Delta)e^{-aT}e^{-a\Delta T}}{2\left(1+e^{-aT}\right)^2}\left\{\frac{(1+w)(1-w)}{\left[w+\left(1-e^{-aT}/1+e^{-aT}\right)\right]^2}\right\}$$

$$+\frac{T^2 e^{-2aT}e^{-a\Delta T}}{\left(1+e^{-aT}\right)^3}\left\{\frac{(1+w)(1-w)^2}{\left[w+\left(1-e^{-aT}/1+e^{-aT}\right)\right]^3}\right\} \qquad (64)$$

In Eqs. 62-64,

T = Sampling interval (sec)

ΔT = Time advance (sec), $0 \leqslant \Delta T < T$, $0 \leqslant \Delta < 1$

The w-plane is related to the z-plane by the bilinear transformation in Eq. 65.

$$w = \frac{z-1}{z+1} \quad , \quad z = \frac{1+w}{1-w} \quad , \quad z = e^{sT} \qquad (65)$$

The corresponding z-plane transforms for the partial fraction expansion terms in Eq. 60 are shown in Eqs. 66-68 (Ref. 2).

$$\left[\frac{e^{\Delta Ts}}{s+a}\right]^T = \frac{e^{-a\Delta T}z}{z-e^{-aT}} \qquad (66)$$

$$\left[\frac{e^{\Delta Ts}}{(s+a)^2}\right]^T = \frac{\Delta T e^{-a\Delta T}z}{z-e^{-aT}} + \frac{Te^{-aT}e^{-a\Delta T}z}{\left(z-e^{-aT}\right)^2} \qquad (67)$$

$$\left[\frac{e^{\Delta Ts}}{(s+a)^3}\right]^T = \frac{(\Delta T)^2 e^{-a\Delta T}z}{2(z-e^{-aT})} + \frac{T^2(1+2\Delta)e^{-aT}e^{-a\Delta T}z}{2\left(z-e^{-aT}\right)^2} + \frac{T^2 e^{-2aT}e^{-a\Delta T}z}{\left(z-e^{-aT}\right)^3} \qquad (68)$$

The s-plane expressions in Eqs. 62-64 and 66-68 represent Laplace transforms of functions advanced in time. For example, $e^{\Delta Ts}/(s + a)$ is the Laplace transform of the continuous time function e^{-at} advanced by ΔT seconds. That is,

$$e^{-a(t+\Delta T)} \quad , \qquad 0 \leq (t+\Delta T) < (t+T) \tag{69}$$

Sampling the advanced time function in Eq. 69 with period T and taking the z-transform results in the advanced z-transform

$$\frac{e^{-a\Delta T}z}{z - e^{-aT}} \tag{70}$$

and the advanced w-transform

$$\frac{e^{-a\Delta T}}{1 + e^{-aT}} \left\{ \frac{w + 1}{w + \left(1 - e^{-aT}/1 + e^{-aT}\right)} \right\} \tag{71}$$

Numerical calculations in DISCRET are carried out in the w-plane to improve the accuracy of the cross-multiplications necessary to form the numerator of the discrete transfer function. The discrete numerator is formed by multiplying each partial fraction expansion numerator by all denominator terms except its own and then summing the resultant products. In general, the poles of the z-transfer function tend to migrate towards the unit circle in the z-plane (i.e., $z \approx 1$). Computationally, severe loss of accuracy can result from the summation of individual partial fraction expansion terms in the z-plane. This inherent inaccuracy can be minimized by performing all possible calculations in the w- or w′-plane where the poles are more reasonably separated. Therefore, the w-plane Eqs. 62-64 are implemented in the computer code. It then becomes a simple task to calculate the corresponding z- and w′-transforms using the bilinear transformations in Eq. 72.

$$w = \frac{z - 1}{z + 1} \quad , \qquad w = \frac{T}{2} w' \tag{72}$$

The s-plane partial fraction expansion terms in Eq. 60 are obtained in the subroutine PARTFR. These are passed to the subroutine WPLN which calculates the w-plane transforms via Eqs. 62-64. The time factor term $e^{\Delta Ts}$ represents a time advance of less than one sampling interval T (i.e., $0 \leq \Delta T < T$). For time delays, an additional delay factor $z^{-1} = 1 - w/1 + w$ is added to Eqs. 62-64 and ΔT is defined by

$$\Delta T = 1 - (D/T)$$

T = Sampling interval (sec)

D = Delay (sec

In this manner, the program can calculate either the advanced or delayed discrete transform. The user inputs a positive time advance (ΔT), a negative time delay (D), or zero for the time increment. From this information alone, the program calculates the advanced, delayed, or standard discrete transform.

The computer code automatically inserts a user-selected data hold (Table 1) into the implementation scheme. The appropriate s-plane zeros and poles associated with each data hold is inserted into the s-plane continuous system G(s) (Eq. 60) by the main program module ADVANZ. The z-plane term $F^T(z)$ in Eq. 60 is added to the transformation by the subroutine WPLN.

D. PROGRAM STRUCTURE

The basic structure of DISCRET consists of the ADVANZ main program and two primary subroutines PARTFR and WPLN. These three program modules along with their supporting subroutines are shown in Fig. 10. The main program ADVANZ first calls PARTFR to obtain the partial fraction expansion terms in the s-plane and then calls WPLN to execute the conversion to the z-, w-, or w′-plane. No external libraries are used with the exception of the normal system routines that support FORTRAN. Parameters are passed between the main program and the two primary

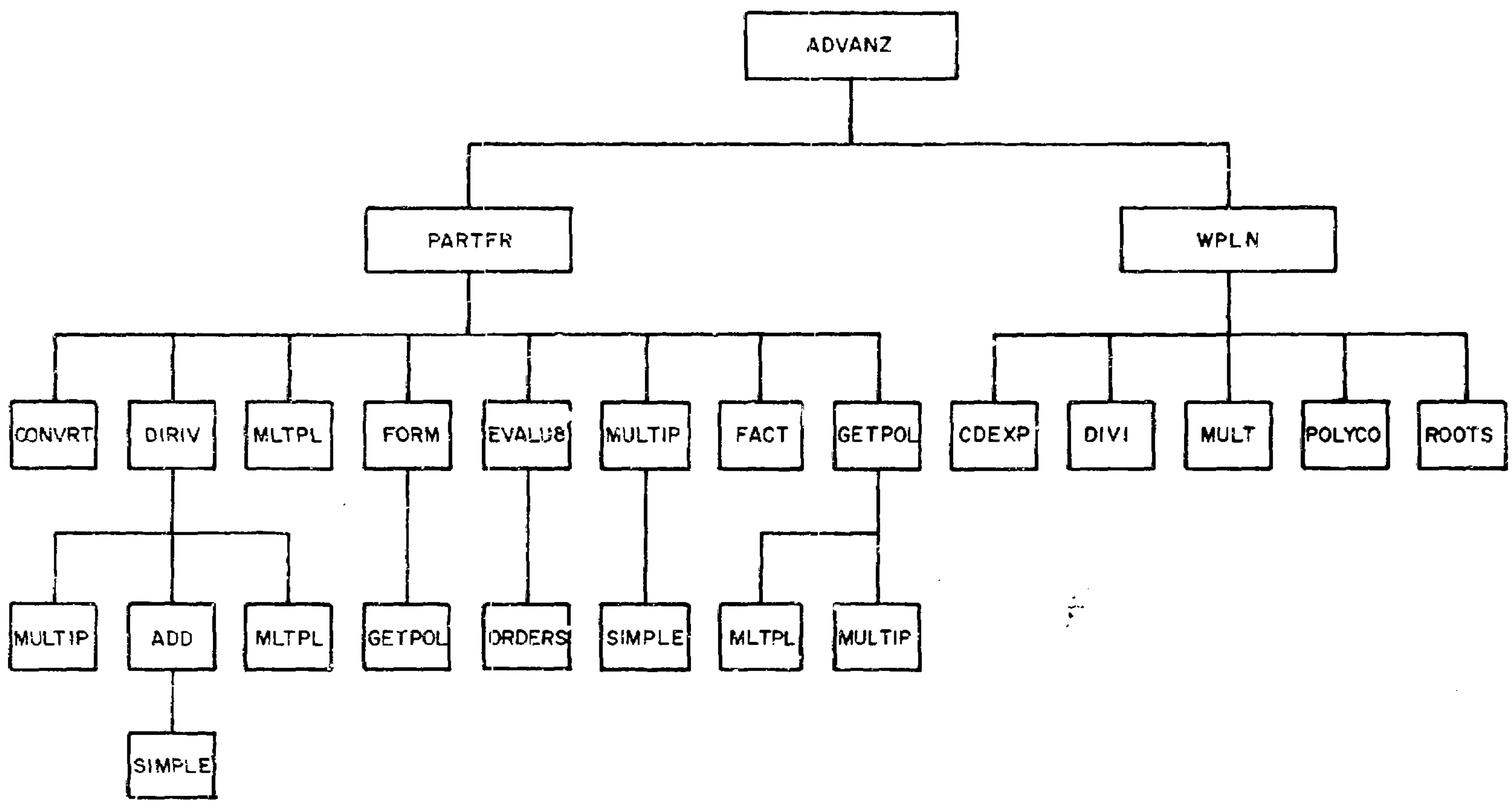


Figure 10. DISCRET Program Structure

subroutines entirely in a COMMON data structure. This lack of formal parameter passing via subroutine arguments was initiated to allow the program to be easily converted to an overlay structure in the TOTAL (Ref. 13) computer program at AFWAL/FIGC. This permits interactive operation of DISCRET as an option in TOTAL. The overlay version at AFFDL transfers the main program functions to the TOTAL main overlay. The main program ADVANZ is then treated as a primary overlay with subroutines PARTFR and WPLN converted to secondary overlays.

The source listing for DISCRET in Volume III is set up in a standard program-subroutine structure (i.e., a main program followed by its subroutines). This listing also includes (in the comment code) the required changes to run the program in an overlay structure. Dividing the program into overlays reduces the amount of computer memory required to execute the program. The overlay code is highlighted with a star (*) character is column one. Removing this code from comment will allow overlay operation. To complete this turnover, the main program card for ADVANZ and the subroutine cards for PARTFR and WPLN must be deleted. In addition, the two call subroutine statements located in ADVANZ must also be deleted.

DISCRET can be run in either a batch or interactive mode. The source code in Volume III is set up for batch operation. The prompting code to run the program in an interactive mode is also included in the comment section of the main program ADVANZ. This code can be identified by the characters "CINT" in the first four columns.

E. DESCRIPTION OF SUBROUTINES

This subsection presents a brief description of the routines used in DISCRET. The general program structure contains a main program ADVANZ, two primary subroutines PARTFR and WPLN, and 16 supporting subroutines. These routines are all coded in FORTRAN.

1. Program ADVANZ

a. Purpose

This is the main program for DISCRET. It reads the input from data cards and transfers it to internal variables and arrays that are located in a common data structure. The program adds appropriate poles and zeros to the input s-plane transfer function according to the data hold selected. It calls PARTFR to calculate the partial fraction expansion terms and then calls WPLN to execute the discrete transformations to the z-, w-, or w′-plane.

b. Input/Output

All data are read from data cards and transferred to PARTFR and WPLN via labeled common.

2. Subroutine PARTFR

a. Purpose

This routine takes the plant description in terms of poles and zeros and outputs the partial fraction expansion coefficients corresponding to each pole. The program drops identical poles and zeros and then calculates the polynomials needed for evaluating the partial fraction expansion terms.

b. Input/Output

All inputs and outputs for this subroutine are handled entirely by common statements.

3. Subroutine WPLN

a. Purpose

This subroutine takes the partial fraction expansion coefficients and the corresponding poles and calculates the w-plane transformation. After determining the w-plane numerator and denominator for each partial

fraction expansion term, the cross product is formed. Each numerator is multiplied by all denominators except its own. The resultant polynomials in w are then summed by adding coefficients for each power of w. The subroutine then calls a root solver to find the zeros of the numerator of the overall transfer function. The roots, poles and w-plane polynomials may then be transformed to the z- or w′-plane.

b. Input/Output

All inputs and outputs for this subroutine are handled entirely by common statements.

4. Subroutine CONVRT (A, NA, PN, NPN)

a. Purpose

The purpose of this routine is to change the format of an array with real polynomial coefficients and corresponding powers of s to allow a place for the imaginary part of the coefficient.

b. Input

1) A: A double precision array with a two place format such that each real coefficient of a polynomial is immediately followed with its corresponding power of s.

2) NA: Number of occupied locations in array A.

c. Output

1) PN: A double precision array with a three place format such that each real coefficient is followed by a zero in the next location and the corresponding power of s in the third location.

2) NPN: The number of occupied locations that are used in PN(3*NA/2).

5. Subroutine DIRIV (PN, NPN, PD, NPD)

a. Purpose

This routine takes the derivative of a transfer function with the numerator polynomial located in PN and the denominator polynomial located in PD. It then stores the numerator of the derivative in PN and the denominator in PD. If

$$P = \sum_{i=0}^{n} A_i s^{n-i}$$

then

$$\frac{d}{ds}(P) = \sum_{i=0}^{n-1} (n - i) A_i s^{n-i-1}$$

and

$$PN = \frac{d}{ds}(PN)*PD - \frac{d}{ds}(PD)*PN$$

$$PD = PD*PD$$

b. Input

1) PN: A double precision array containing the numerator polynomial in the format: real part, imaginary part, and the order of s stored in back-to-back locations.

2) NPN: Number of occupied elements in array PN.

3) PD: A double precision array containing the denominator polynomial in the same format as the numerator polynomial.

4) NPD: Number of occupied elements in array PD.

c. Output

1) PN: Numerator polynomial (in the same format as the input) of the derivative function.

2) NPN: Number of occupied elements in PN.

3) PD: Denominator polynomial of the derivative function.

4) NPD: Number of occupied elements in PD.

6. Subroutine MLTPL (C, N, D, E, M)

a. Purpose

This routine multiplies the real polynomial coefficients by a scale factor and stores the resultant polynomial in a new array.

b. Input

1) C: A Double precision array containing polynomial coefficients and corresponding powers of s.

2) N: Number of occupied elements in array C.

3) D: Scale factor which multiplies all odd locations of array C.

c. Output

1) E: A double precision array containing scaled polynomial coefficients.

2) M: Number of occupied elements in array E.

7. Subroutine FORM (RD, NRD, MULT, NEGLCT, P2, NP2)

a. Purpose

The purpose of this routine is to form the denominator polynomial that will be used to evaluate the partial fraction expansion coefficient corresponding to a pole of the plant. The routine multiplies all of the

poles together, excluding the pole (and its conjugate if the pole has an imaginary part) for which the partial fraction expansion coefficient is being sought.

b. Input

1) RD: A double precision array containing the poles of the plant, real part then imaginary part.

2) NRD:ɃɃɃ: Two times the number of distinct poles of the plant (number of locations used in array RD).

3) MULT: Array contains multiplicity corresponding to each pole contained in array RD (NRD/2 locations).

4) NEGLCT: An integer array containing all zeros except in the location corresponding to the pole for which the partial fraction expansion coefficient is being determined, where a one appears. If the pole is complex, a one also appears in location corresponding to the conjugate of the pole (NRD/2 locations).

c. Output

1) P2:ɃɃɃ: A double precision array containing a polynomial representing the product of all the poles except the one for which the partial fraction expansion coefficient is being sought (and its conjugate if complex). The odd locations contain the coefficients of the polynomial, and the even locations contain the corresponding power of s. All coefficients are real.

2) NP2:ɃɃɃ: Number of occupied locations in the P2 array.

8. <u>Subroutine EVALU8 (P, NP, R, V, ZF)</u>

a. Purpose

This routine evaluates a polynomial at the pole for which the partial fraction expansion coefficient is being sought.

b. Input

1) P: A double precision array containing the polynomial coefficients in descending order with the coefficient (real part, imaginary part) and power of s stored in separate back-to-back locations.

2) NP: Number of occupied locations in array P.

3) R: A double precision array containing the real part of the pole to be evaluated in the first location and the imaginary part in the next location.

c. Output

1) V: A double precision array containing the value of the polynomial after the pole of interest is evaluated. The real part is in the first location and the imaginary part in the second.

2) ZF: A scale factor that is used in the routine to maintain numerical accuracy for large products.

9. <u>Subroutine MULTIP (C1, NT1, C2, NT2, C3, NT3, N)</u>

a. Purpose

This routine multiplies two polynomials and then calls the subroutine SIMPLE to combine the coefficients with like powers of s.

b. Input

1) N: An integer that specifies which format the polynomials are in. A two corresponds to real coefficients with two locations necessary for each polynomial term. A three corresponds to real and imaginary coefficients with three locations necessary for each polynomial term.

2) C1: A double precision array containing a polynomial in the format specified by N.

3) NT1: Number of occupied elements in array C1.

4) C2: A double precision array containing a second polynomial in the format specified by N.

5) NT2: Number of occupied elements in array C2.

c. Output

1) C3: A double precision array containing the product of the polynomials C1 and C2 in the format specified by N.

2) NT3:ƀƀƀ: Number of occupied elements in array C3.

10. Subroutine GETPOL (NR, NRN, A, NA)

a. Purpose

This routine takes a set of roots and multiplies them to form a polynomial.

b. Input

1) RN: A double precision array containing the roots to be multiplied together. The roots are stored real part, then imaginary part. In location RN(2*NRN+1), a scale factor is stored that multiplies the polynomial.

2) NRN: Number of roots to be multiplied together.

b. Output

1) A: A double precision array with coefficients of the polynomial times the scale factor in the odd locations in descending order and corresponding powers of s located in the even locations.

2) NA: Number of locations used in array A.

11. Subroutine ADD (C1, NT1, C2, NT2, C3, NT3, M)

a. Purpose

The routine adds polynomials C1 and C2 together and places the sum in array C3. The polynomials are first placed sequentially in array C3

and then the subroutine SIMPLE is called to add coefficients of like powers of s. The dimension of array C3 must equal the combined dimensions of C1 and C2.

b. Input

1) M: An integer constant specifying, as in MULTIP, which format the polynomial coefficients are in.

2) C1: A double precision array containing polynomial coefficients and corresponding powers of s in the format specified by M.

3) NT1: Number of occupied elements in array C1.

4) C2: A double precision array containing a second polynomial in the format specified by M.

5) NT2: Number of occupied elements in array C2.

c. Outputs

1) C3: A double precision array containing a polynomial which is the sum of the polynomials C1 and C2.

2) NT3: Number of occupied elements in array C3.

12. <u>Subroutine SIMPLE (P, N, K)</u>

a. Purpose

This routine combines the coefficients of a polynomial into the least number of coefficients by adding together the coefficients with like powers of s.

b. Input

1) K: An integer constant specifying, as in MULTIP, the format of the polynomial P.

2) P: A double precision array containing a polynomial in the format specified by K.

3) N: Number of occupied elements in array P.

c. Output

1) P: Array containing least number of coefficients necessary to specify the polynomial read in.

2) N: Number of occupied elements in the output array P.

13. <u>Subroutine ORDER3 (P, NP, K)</u>

a. Purpose

This routine orders the polynomial coefficients into descending powers of s.

b. Input

1) K: An integer constant specifying, as in MULTIP, the format of the input polynomial P.

2) P: A double precision array containing the input polynomial.

3) NP: Number of occupied locations in array P.

c. Output

1) P: Array containing the polynomial in descending powers of s.

2) NP: Number of occupied locations in output array P.

14. Subroutine CDEXP (A, B, X, Y)

a. Purpose

This routine calculates the exponential function of a complex number in double precision.

b. Input

1) A: Real part (double precision) of argument of exponential function.

2) B: Imaginary part (double precision) of argument of exponential function.

c. Output

1) X: Real part (double precision) of exponential function.

2) Y: Imaginary part (double precision) of exponential function.

15. Subroutine MULT (A, B, C, D, X, Y)

a. Purpose

This routine multiplies a double precision complex number by a double precision complex number.

b. Input

1) A, B: Real and imaginary parts of the first number.

2) C, D: Real and imaginary parts of second number.

c. Output

1) X: Real part of the complex product.

2) Y: Imaginary part of the complex product.

16. Subroutine DIVI (A, B, C, D, X, Y)

a. Purpose

This routine divides a double precision complex number by a double precision complex number.

b. Input

1) A, B: Real and imaginary parts of the first number.

2) C, D: Real and imaginary parts of second number.

c. Output

1) X: Real part of the complex division.

2) Y: Imaginary part of the complex division.

17. Function Fact (n)

This is a function subroutine that calculates n!.

18. Subroutine POLYCO (A, B, RR, RI, N)

a. Purpose

The purpose of this routine is to form a polynomial from a set of roots. Both real and imaginary coefficients are calculated.

b. Input

1) RR: Double precision array containing the real part of each root.

2) RI: Double precision array containing the imaginary part of each root.

3) N: Number of input roots.

c. Output

1) A: Double precision array containing the real coefficients of the polynomial.

2) B: Double precision array containing the imaginary part of the polynomial coefficients.

19. Subroutine ROOTS (A, B, NN, RR, RI)

a. Purpose

This subroutine finds the roots of a polynomial with complex coefficients.

b. Input

1) A: Double precision array containing the real part of the polynomial coefficients in descending powers.

2) Double precision array containing the imaginary part of the polynomial coefficients in descending powers.

3) NN: Order of input polynomial.

c. Output

1) RR: Double precision array containing the real part of each root.

2) RI: Double precision array containing the imaginary part of each root.

F. PROGRAM VARIABLES WITHIN LABELED COMMON

Two labeled COMMON blocks are used in the main program ADVANZ and the two primary subroutines PARTFR and WPLN. Variables and arrays in the main program and the two primary subroutines share the same storage locations by means of the COMMON statement. These variables and arrays are stored in the order in which they appear in the block specification.

The COMMON blocks replace the subroutine arguments in PARTFR and WPLN. This arrangement allows DISCRET to be easily converted to an overlay structure. The variables and arrays located within these COMMON data blocks are outlined below. Each variable and array is listed individually with a brief description of its purpose.

1. COMMON/ADVCZ/RN,RD,NRTN,NRTD,MM,VR,VI,BR,BI,T,NH,AM,AO,TXFORM,CPLR

Variable	Purpose
RN	A double precision array containing the zeros of the continuous s-plane transfer function G(s). The zeros are stored real part then imaginary part (required storage equals two times number of zeros).
RD	A double precision array containing the poles of G(s) in the same format as the zeros.
NRTN	Number of zeros in G(s).
NRTD	Number of poles in G(s).
MM	Integer array containing multiplicity of poles corresponding to the partial fraction expansion coefficients located in arrays VR and VI.
VR,VI	Double precision arrays containing partial fraction expansion coefficients (real and imaginary) for the poles of G(s) including those poles introduced by the data hold.
BR,BI	Double precision arrays containing corresponding poles (real and imaginary) for the VR and VI arrays.
T	Sampling time (sec).
NH	Integer variables used in the data hold option.
AO	Gain of the G(s) transfer function.
TXFORM	Transform option variable - Z, W, or WP.
CPLR	Data hold option variable - NON, ZOH, 1ST, 2ND, or SLE.

2. COMMON/TOTL12/CLNPOLY(51),CLDPOLY(51),CLZERO(50,2), CLPOLE(50,2),NCLZ,NCLP,CLK,CLNK,CLDK

CLNPOLY — Single precision array containing the numerator polynomial coefficients for the discrete transfer function G(z), G(w), or G(w′). The coefficients are stored sequentially back-to-back with the highest order coefficient first in the array.

CLDPOLY — Single precision array containing the denominator polynomial coefficients for the discrete transfer function in the same format as the numerator array.

CLZERO — Single precision array containing the zeros of the discrete transfer function. The real part of the nth zero is stored in the first column (n,1) and the imaginary part in the second (n,2).

CLPOLE — Single precision array containing the poles of the discrete transfer function in the same format as the zeros.

NCLZ — Number of zeros in the discrete transfer function.

NCLP — Number of poles in the discrete transfer function.

CLK — Total gain for the discrete transfer function (CLK = CLNK/CLDK).

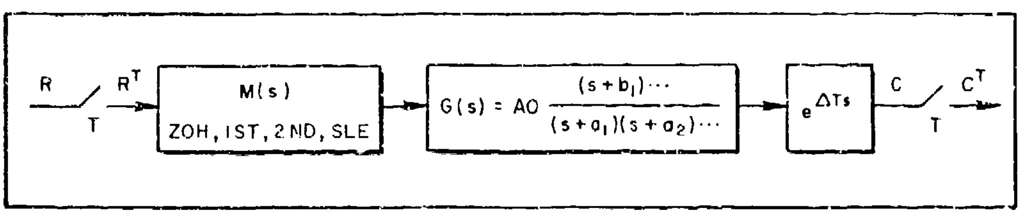
CLNK Numerator gain for the discrete transfer function.

CLDK Denominator gain for the discrete transfer function.

G. PROGRAM OPERATING INSTRUCTIONS

The example in Fig. 11 will be used to illustrate the input and output data structure for the DISCRET computer program. Input data items are free-form (free-format) with separators rather than in fixed-size fields. The two exceptions are the alphanumeric inputs which select the desired data hold and discrete transform. The free-format input data consist of a string of values separated by one or more blanks, or by a comma or slash, either of which may be preceded or followed by any number of blanks. A line boundary, such as an end of record or end of card, also serves as a value separator (Ref. 14).

The input is divided into three main blocks of data. The first block contains the basic parameters that define the s-plane system G(s) to be transformed. These data are placed on the first data card in a free format. The alphanumeric code for the data hold and type of discrete transform are inserted on data cards two and three in an A3 and A2 format, respectively. The final block of data is again free format.



$$\frac{C^T}{R^T} = \left[e^{\Delta Ts} G(s)M(s)\right]^T$$

Figure 11. Sampled Continuous System

This input starts on card four and consists of the zeros and poles of the continuous s-plane transfer function G(s). The required input data are outlined in Table 2.

The arithmetic sign (i.e., positive or negative) of the time increment ΔT selects the advanced (positive ΔT) or delayed (negative ΔT) discrete transform. For example, for a sampling period of T = 1.0 and a time increment of 0.3, the advanced discrete transform is specified as $\Delta T = 0.3$ and the delayed discrete transform as $\Delta T = -0.3$. For the standard discrete transform the time increment is $\Delta T = 0.0$.

TABLE 2. INPUT DATA FOR DISCRET

VARIABLE	PURPOSE
Data Card One - Free-Format	
NRTN	Number of zeros in G(s)
M	Number of poles in G(s)
T	Sampling time (sec)
AO	G(s) gain
AM, (ΔT)	Time increment option: ΔT, $-\Delta T$, or zero (sec)
Data Cards Two and Three - A3, A2 Format	
CPLR	Data hold option: NON, ZOH, 1ST, 2ND, or SLE
TXFORM	Transform option: Z, W, or WP
Data Cards Four to nth - Free-Format	
RN(2*NRTN)	Zeros of G(s): real part then imaginary part
RD(2*M)	Poles of G(s): real part then imaginary part

The data hold and transform options are input in a coded alphanumeric format. These options with their respective alphanumeric codes are given in Table 3. The alphanumeric input code for the data hold is placed in Columns 1-3 on data card two (left justified). The discrete transform code appears on card three in Columns 1-2 (left justified).

TABLE 3. ALPHANUMERIC CODES FOR DISCRET

Data Hold Option	Input Code
None	NON
Zero order hold	ZOH
First order hold	1ST
Second order hold	2ND
Slewer	SLE
Transform Option	Input Code
z transform	Z
w transform	W
w′ transform	WP

The order of the G(s) transfer function must be equal to or less than 50th. Pole multiplicity up to and including three is permitted. There is no restriction on the number of sets of repeated poles. The zeros and poles of G(s) are input sequentially in a free-format (starting on data card four) on as many data cards as is necessary. The real and imaginary parts are separated with a valid separator (i.e., a comma, a slash, or one or more blanks). The zeros are given first followed by the poles. For real roots, 0.0 must be input for the imaginary part.

The output data from DISCRET can be divided into two main sections. The first section deals with the input parameters for the s-plane continuous system G(s) and those parameters that define the discrete

transformation. The second section contains the results of the transformation process —— the discrete transfer function G(z), G(w), or G(w′). The program first prints the transform options that have been selected. This includes the sampling time, data hold, type of discrete transform, and the time increment. This is followed with a list of the s-plane zeros and poles for G(s) including those introduced by the data hold. The partial fraction expansion coefficients for G(s) and its data hold are printed next. The program then outputs the numerator and denominator polynomials and the zeros and poles for the discrete transfer function.

Table 4 contains nine sets of input data in card image format. The sampled s-plane systems for these examples are depicted in Fig. 12. The discrete transfer functions for each of these systems are given in Eqs. 73-75.

$$\frac{C^T}{R^T} = \left[\frac{5s}{(s + 1 - 2j)(s + 1 + 2j)} \; \frac{1 - e^{-sT}}{s}\right]^T \tag{73}$$

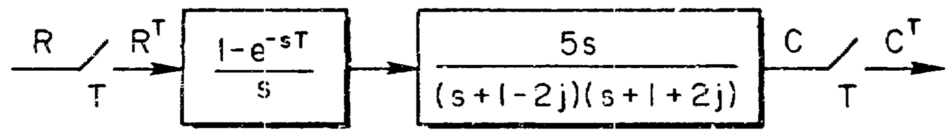
$$\frac{C^T}{R^T} = \left[\frac{e^{.004s}(s + .03)(s + 6.3)(s - 6)}{(s + 2)(s - 1.2)(s + .01 - .07j)(s + .01 + .07j)} \; \frac{1 - e^{-sT}}{s}\right]^T \tag{74}$$

$$\frac{C^T}{R^T} = \left[\frac{e^{-.004s}(s + .03)(s + 6.3)(s - 6)}{(s + 2)(s - 1.2)(s + .01 - .07j)(s + .01 + .07j)} \; \frac{(1 - e^{-sT})^2}{Ts^2}\right]^T \tag{75}$$

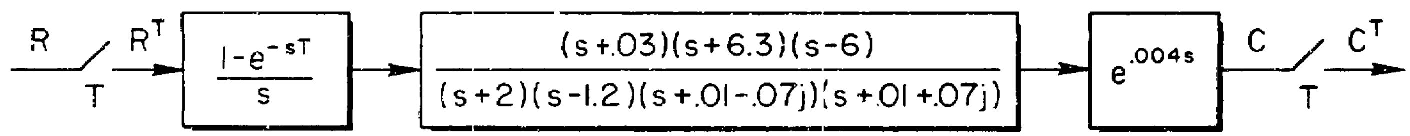
These transfer functions are calculated by the DISCRET computer program.

The output for each data set in Table 4 is shown in Figs. 13-21. The first three sets of data calculate the standard z-, w-, and w′- discrete transforms using a zero order hold and a sampling period of T = 0.1 (i.e., TXFORM = Z, W, and WP; CPLR = ZOH; and ΔT = 0.0). The output for these three examples is shown in Figs. 13-15. In data

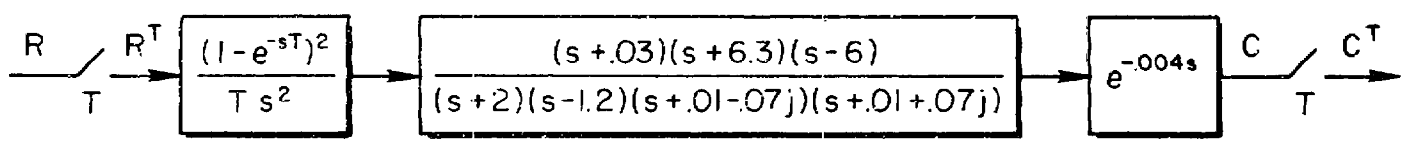
sets 4-6, the advanced z-, w-, and w'-transforms are calculated for the second system given by Eq. 74. A ZOH is used with a sampling period of $T = 0.04$ and a time advance of $\Delta T = 0.004$ seconds. These advanced discrete transfer functions appear in Figs. 16-18. The last three sets of data use the same s-plane system G(s) with a slewer data hold and a time delay of $\Delta T = -0.004$ (Eq. 75). Figures 19-21 contain the delayed discrete transforms for these inputs.



Data Sets 1-3



Data Sets 4-6



Data Sets 7-9

Figure 12. Sampled Systems for Table 4 Input Data

TABLE 4. EXAMPLES OF INPUT DATA FOR DISCRET

CARDS	DATA SET 1	CARDS	DATA SET 2	CARDS	DATA SET 3
1	1,2,0.1,5.0,0.0	1	1,2,0.1,5.0,0.0	1	1,2,0.1,5.0,0.0
2	ZOH	2	ZOH	2	ZOH
3	Z	3	W	3	WP
4	0.0,0.0,-1.0,2.0	4	0.0,0.0,-1.0,2.0	4	0.0,0.0,-1.0,2.0
5	-1.0,-2.0	5	-1.0,-2.0	5	-1.0,-2.0
CARDS	DATA SET 4	CARDS	DATA SET 5	CARDS	DATA SET 6
1	3,4,.04,-0.17,.004	1	3,4,.04,-.017,.004	1	3,4,.04,-.017,.004
2	ZOH	2	ZOH	2	ZOH
3	Z	3	W	3	WP
4	-.3,0.,6.,0.,-6.3	4	-.3,0.,6.,0.,-6.3	4	-.3,0.,6.,0.,-6.3
5	0.,1.2,0.,-2.,0.	5	0.,1.2,0.,-2.,0.	5	0.,1.2,0.,-2.,0.
6	-.01,.07,-.01,-.07	6	-.01,.07,-.01,-.07	6	-.01,.07,-.01,-.07
CARDS	DATA SET 7	CARDS	DATA SET 8	CARDS	DATA SET 9
1	3,4,.04,-.017,-.004	1	3,4,.04,-.017,-.004	1	3,4,.04,-.017,-.004
2	SLE	2	SLE	2	SLE
3	Z	3	W	3	WP
4	-.3,0.,6.,0.,-6.3	4	-.3,0.,6.,0.,-6.3	4	-.3,0.,6.,0.,-6.3
5	0.,1.2,0.,-2.,0.	5	0.,1.2,0.,-2.,0.	5	0.,1.2,0.,-2.,0.
6	-.01,.07,-.01,-.07	6	-.01,.07,-.01,-.07	6	-.01,.07,-.01,-.07

```
TSAMP (SEC)=.1   DELTA (SEC)=0.
HOLD OPTION = ZOH   TRANSFORM OPTION = Z
GAIN =   .5000000000D+01



ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERATOR

RE  0.                IM


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINATOR

RE  -.10000000D+01   IM   .20000000D+01 RE  -.10000000D+01   IM  -.20000000D+01 RE


PARTIAL FRACTIONS
             N U M E R A T O R                          D E N O M I N A T O R
    REAL PART              IMAG. PART              REAL PART         IMAG. PART          EXPONENT
    0.                   -1.25000000D+00        -1.00000000D+00    2.00000000D+00          1
    0.                    1.25000000D+00        -1.00000000D+00   -2.00000000D+00          1
```

Figure 13a. DISCRET Output for Data Set 1

```
NUMERATOR ROOTS IN  Z

(   1.000000000     ,  0.                 )

DENOMINATOR ROOTS IN Z

(   .8868009118     ,   .1797634443     )
(   .8868009118     ;  -.1797634443     )

COEFFICIENTS OF NUMERATOR IN  Z

(   1.000000000     ) Z**    1
(  -1.000000000     ) Z**    0

COEFFICIENTS OF DENOMINATOR IN Z

(   1.000000000     ) Z**    2
(  -1.773601824     ) Z**    1
(    .8187307531    ) Z**    0

CLK=     .4494086108
CLKK=     .4494086108
CLKK=    1.000000000
```

Figure 13b. DISCRET Output for Data Set 1

$$\frac{C^T}{R^T} = \frac{C(z)}{R(z)} = \frac{(.4494086108)z - (.4494086108)}{z^2 - (1.773601824)z + (.8187307531)} \quad , \quad z = e^{sT} = e^{s(.1)}$$

$$= \frac{(.4494086108)\,(z - 1.0)}{(z - .8868009118 + j.1797634443)\,(z - .8868009118 - j.1797634443)}$$

Figure 13c. DISCRET Output for Data Set 1

```
TSAMP (SEC)=.1   DELTA (SEC)=0.
HOLD OPTION = ZOH   TRANSFORM OPTION = W
GAIN =    .5000000000D+01


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERATOR

RE  0.               IM


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINATOR

RE  -.10000000D+01   IM   .20000000D+01 RE  -.10000000D+01   IM  -.20000000D+01 RE


PARTIAL FRACTIONS
              N U M E R A T O R                           D E N O M I N A T O R
         REAL PART               IMAG. PART             REAL PART          IMAG. PART         EXPONENT
        0.                    -1.25000000D+00         -1.00000000D+00     2.00000000D+00         1
        0.                     1.25000000D+00         -1.00000000D+00    -2.00000000D+00         1
```

Figure 14a. DISCRET Output for Data Set 2

```
COEFFICIENTS OF NUMERATOR IN  W

  (   1.000000000     )W**  2
  (  -1.000000000     )W**  1
  (  0.               )W**  0


           NUMERATOR ROOTS IN  W

(   1.000000000    ,  0.              )
(  0.              ,  0.              )


          DENOMINATOR ROOTS IN W

(  -.5046004039E-01,   .1000817382    )
(  -.5046004039E-01,  -.1000817382    )


COEFFICIENTS OF DENOMINATOR IN W

  (   1.000000000     )W**  2
  (   .1009200808     )W**  1
  (   .1256257001E-01 )W**  0


  CLK=    -.2502043456
  CLNK=    -.2502043456
  CLDK=     1.000000000
```

Figure 14b. DISCRET Output for Data Set 2

$$\frac{C^T}{R^T} = \frac{C(w)}{R(w)} = \frac{(-.2502043456)w^2 + (.2502043456)w}{w^2 + (.1009200808)w + (.01256257001)} \quad , \quad w = \frac{z-1}{z+1} , \quad z = e^{s(.1)}$$

$$= \frac{(-.2502043456)\ w(w - 1.0)}{(w + .05046004039 + j.1000817382)\ (w + .05046004039 - j.1000817382)}$$

Figure 14c. DISCRET Output for Data Set 2

```
TSAMP (SEC)=.1   DELTA (SEC)=0.
HOLD OPTION = ZOH   TRANSFORM OPTION = WP
GAIN =    .50000000000D+01



ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERATOR

RE  0.                IM


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINATOR

RE  -.10000000D+01   IM   .20000000D+01 RE  -.10000000D+01   IM  -.20000000D+01 RE


PARTIAL FRACTIONS
                N U M E R A T O R                          D E N O M I N A T O R
          REAL PART          IMAG. PART               REAL PART        IMAG. PART        EXPONENT
        0.                 -1.25000000D+00           -1.00000000D+00   2.00000000D+00       1
        0.                  1.25000000D+00           -1.00000000D+00  -2.00000000D+00       1
```

Figure 15a. DISCRET Output for Data Set 3

```
COEFFICIENTS OF NUMERATOR IN  W

 (   1.000000000     )W**  2
 (  -20.00000000     )W**  1
 (  0.               )W**  0


          NUMERATOR ROOTS IN  W

(   20.00000000     ,  0.                )
(  0.               ,  0.                )


          DEMONINATOR ROOTS IN W

(  -1.009200808     ,   2.001634765     )
(  -1.009200808     ,  -2.001634765     )


 COEFFICIENTS OF DENOMINATOR IN W

 (   1.000000000     )W**  2
 (   2.018401616     )W**  1
 (   5.025020003     )W**  0


 CLK=    -.2502043456
 CLNK=    -.6255100641E-03
 CLDK=     .2500000000E-02
```

Figure 15b. DISCRET Output for Data Set 3

$$\frac{C^T}{R^T} = \frac{C(w')}{R(w')} = \frac{(-.2502043456)w'^2 + (5.00408690)w'}{w'^2 + (2.018401616)w' + (5.025028003)} \quad , \quad w' = \frac{2}{T}\frac{z-1}{z+1} \, , \quad z = e^{s(.1)}$$

$$= \frac{(-.2502043456)\, w'(w' - 20.0)}{(w' + 1.009200808 + j2.001634765)\,(w' + 1.009200808 - j2.001634765)}$$

Figure 15c. DISCRET Output for Data Set [illegible]

```
TSAMP (SEC)=.04   DELTA (SEC)=.004
HOLD OPTION = ZOH   TRANSFORM OPTION = Z
GAIN =  -.17000000000D-01


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERATOR

RE  -.30000000D+00   IM  0.          RE   .60000000D+01   IM  0.          RE  -.63000000D+01   IM  0.


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINATOR

RE   .12000000D+01   IM  0.          RE  -.20000000D+01   IM  0.          RE  -.10000000D-01   IM   .70000000D-01
RE  -.10000000D-01   IM  -.70000000D-01 RE  0.            IM  0.          RE

PARTIAL FRACTIONS

              N U M E R A T O R                        D E N O M I N A T O R
        REAL PART          IMAG. PART             REAL PART          IMAG. PART           EXPONENT
     1.62738257D-01       0.                   1.20000000D+00       0.                      1
    -3.91771753D-02       0.                  -2.00000000D+00       0.                      1
     7.97071946D+00       9.373276440D-01     -1.00000000D-02       7.00000000D-02          1
     7.97071946D+00      -9.373276440D-01     -1.00000000D-02      -7.00000000D-02          1
    -1.60650000D+01       0.                   0.                   0.                      1
```

Figure 16a. DISCRET Output for Data Set 4

```
                 NUMERATOR ROOTS IN  Z

(    1.271558504    ,  0.                )
(     .9880717085   ,  0.                )
(     .7770207068   ,  0.                )
(  -8.893580068     ,  0.                )


                DENOMINATOR ROOTS IN Z

(    1.049170655    ,  0.                )
(     .9231163464   ,  0.                )
(     .9995961616   ,    .2798876567E-02 )
(     .9995961616   ,   -.2798876567E-02 )


          COEFFICIENTS OF NUMERATOR IN  Z

   (    1.000000000          ) Z**     4
   (    5.856929150          ) Z**     3
   (  -23.99452764           ) Z**     2
   (   25.81273725           ) Z**     1
   (   -8.682284702          ) Z**     0


          COEFFICIENTS OF DENOMINATOR IN Z

   (    1.000000000          ) Z**     4
   (   -3.971479325          ) Z**     3
   (    5.910687935          ) Z**     2
   (   -3.206940727          ) Z**     1
   (     .9677320067         ) Z**     0


   CLK=     -.6796370909E-04
   CLMK=     -.6796370909E-04
   CLDK=      1.000000000
```

Figure 16b. DISCRET Output for Data Set 4

```
TSAMP (SEC)=.04   DELTA (SEC)=.004
HOLD OPTION = ZOH   TRANSFORM OPTION = W
GAIN =   -.17000000000D-01


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERATOR

RE  -.30000000D+00   IM  0.          RE   .60000000D+01   IM  0.          RE  -.63000000D+01   IM  0.


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINATOR

RE   .12000000D+01   IM  0.          RE  -.20000000D+01   IM  0.          RE  -.10000000D-01   IM   .70000000D-01
RE  -.10000000D-01   IM  -.70000000D-01 RE  0.             IM  0.          RE


PARTIAL FRACTIONS

           N U M E R A T O R                            D E N O M I N A T O R
     REAL PART            IMAG. PART              REAL PART            IMAG. PART           EXPONENT
   1.62738257D-01        0.                     1.20000000D+00        0.                      1
  -3.91771753D-02        0.                    -2.00000000D+00        0.                      1
   7.97071946D+00        9.37327644D-01        -1.00000000D-02        7.00000000D-02          1
   7.97071946D+00       -9.37327644D-01        -1.00000000D-02       -7.00000000D-02          1
  -1.60650000D+01        0.                     0.                    0.                      1
```

Figure 17a. DISCRET Output for Data Set 5

```
COEFFICIENTS OF NUMERATOR IN  W

(    1.000000000      )W**  4
(   -1.241438460      )W**  3
(   -.2992032247E-01  )W**  2
(    .1866682115E-01  )W**  1
(    .1128072236E-03  )W**  0


NUMERATOR ROOTS IN  W

(    .1195472198   ,  0.                 )
(   -.5999930219E-02, 0.                 )
(   -.1254792880   ,  0.                 )
(   1.253370458    ,  0.                 )


DENOMINATOR ROOTS IN W

(    .23995393906E-01,  0.               )
(   -.3997668031E-01,   0.               )
(   -.2000003893E-03,    .1400000859E-02 )
(   -.2000003893E-03,   -.1400000859E-02 )


COEFFICIENTS OF DENOMINATOR IN W

(    1.000000000      )W**  4
(    .1638328803E-01  )W**  3
(   -.9509108182E-03  )W**  2
(   -.35175.7908E-06  )W**  1
(   -.19186.0752E-08  )W**  0


CLK=     .2732312845E-03
CLNK=     .2732312845E-03
CLDK=    1.000000000
```

Figure 17b. DISCRET Output for Data Set 5

```
TSAMP (SEC)=.04   DELTA (SEC)=.004
HOLD OPTION = ZOH   TRANSFORM OPTION = WP
GAIN =  -.1700000000D-01


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERATOR

RE  -.30000000D+00   IM  0.            RE   .64000000D+01   IM  0.            RE  -.63000000D+01   IM  0.


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINATOR

RE   .12000000D+01   IM  0.            RE  -.20000000D+01   IM  0.            RE  -.10000000D-01   IM   .70000000D-01
RE  -.10000000D-01   IM  -.70000000D-01 RE  0.              IM  0.            RE


PARTIAL FRACTIONS

          N U M E R A T O R                          D E N O M I N A T O R
      REAL PART          IMAG. PART              REAL PART          IMAG. PART          EXPONENT
   1.627382570D-01      0.                   1.200000000D+00      0.                      1
  -3.917717530D-02      0.                  -2.000000000D+00      0.                      1
   7.970719460D+00      9.373276410D-01     -1.000000000D-02      7.000000000D-02         1
   7.970719460D+00     -9.373276410D-01     -1.000000000D-02     -7.000000000D-02         1
  -1.606500000D+01      0.                   0.                   0.                      1
```

Figure 18a. DISCRET Output for Data Set 6

```
COEFFICIENTS OF NUMERATOR IN  W

(    1.000000000     )W**  4
(   -62.07192299     )W**  3
(   -74.80080618     )W**  2
(    2333.352644     )W**  1
(    705.0455850     )W**  0


         NUMERATOR ROOTS IN  W

(    5.977360988     ,  0.                   )
(   -.2999965110     ,  0.                   )
(   -6.273964402     ,  0.                   )
(    62.66852292     ,  0.                   )


         DENOMINATOR ROOTS IN W

(    1.199769653     ,  0.                   )
(   -1.998934016     ,  0.                   )
(   -.1000001947E-01,    .7000004293E-01)
(   -.1000001947E-01,   -.7000004293E-01)


COEFFICIENTS OF DENOMINATOR IN W

(    1.000000000     )W**  4
(    .8191644014     )W**  3
(   -2.377277046     )W**  2
(   -.4396947385E-01 )W**  1
(   -.1199131720E-01 )W**  0


CLK=      .2732312845E-03
CLNK=     .4371700553E-10
CLDK=     .1600000000E-06
```

Figure 18b. DISCRET Output for Data Set 5

```
TSAMP (SEC)=.04   DELTA (SEC)=-.004
HOLD OPTION = SLE   TRANSFORM OPTION = Z
GAIN =  -.17000000D-01



ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERATOR

RE  -.30000000D+00   IM  0.          RE   .60000000D+01   IM  0.          RE  -.63000000D+01   IM  0.


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINATOR

RE   .12000000D+01   IM  0.          RE  -.20000000D+01   IM  0.          RE  -.10000000D-01   IM   .70000000D-01
RE  -.10000000D-01   IM  -.70000000D-01 RE  0.            IM  0.          RE


PARTIAL FRACTIONS

            N U M E R A T O R                          D E N O M I N A T O R
       REAL PART          IMAG. PART             REAL PART         IMAG. PART          EXPONENT
     1.35615214D-01      0.                     1.20000000D+00     0.                      1
     1.05885876D-02      0.                    -2.00000000D+00     0.                      1
    -2.81885190D+00     -1.13464728D+02        -1.00000000D-02     7.00000000D-02          1
    -2.81885190D+00      1.13464728D+02        -1.00000000D-02    -7.00000000D-02          1
    -1.60650000D+01      0.                     0.                 0.                      2
     5.49250000D+00      0.                     0.                 0.                      1
```

Figure 19a. DISCRET Output for Data Set 7

```
                 NUMERATOR ROOTS IN  Z

(    1.271230250     ,  0.                  )
(     .9880717129    ,  0.                  )
(     .7772355278    ,  0.                  )
(   -.8545112596E-02,  0.                  )
(   -1.441750048     ,  0.                  )


                DENOMINATOR ROOTS IN Z

(    1.049170655      ,  0.                 )
(     .9231163464     ,  0.                 )
(     .9995961616     ,   .2798876567E-02   )
(     .9995961616     ,  -.2798876567E-02   )
(   0.                ,  0.                 )
(   0.                ,  0.                 )


        COEFFICIENTS OF NUMERATOR IN  Z

   (    1.000000000      ) Z**     5
   (   -1.586250330      ) Z**     4
   (   -1.379476787      ) Z**     3
   (    3.354744558      ) Z**     2
   (   -1.378764818      ) Z**     1
   (   -.1202751276E-01  ) Z**     0


        COEFFICIENTS OF DENOMINATOR IN Z

   (    1.000000000      ) Z**     6
   (   -3.971479325      ) Z**     5
   (    5.910687936      ) Z**     4
   (   -3.906940727      ) Z**     3
   (     .9677320867     ) Z**     2
   (   0.                ) Z**     1
   (   0.                ) Z**     0


   CLK=     -.2736309695E-03
   CLNK=     -.2736309695E-03
   CLDK=      1.000000000
```

Figure 19b. DISCRET Output for Data Set 7

```
TSAMP (SEC)=.04   DELTA (SEC)=-.004
HOLD OPTION = SLE   TRANSFORM OPTION = W
GAIN =  -.1700000000D-01



ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERATOR

RE  -.30000000D+00   IM  0.        RE  .60000000D+01   IM  0.        RE  -.63000000D+01   IM  0.


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINATOR

RE   .12000000D+01   IM  0.        RE  -.20000000D+01   IM  0.
RE  -.10000000D-01   IM  -.70000000D-01 RE  0.        IM  0.        RE  -.10000000D-01   IM  .70000000D-01
                                                                    RE
```

PARTIAL FRACTIONS

NUMERATOR REAL PART	NUMERATOR IMAG. PART	DENOMINATOR REAL PART	DENOMINATOR IMAG. PART	EXPONENT
1.35615214D-01	0.	1.20000000D+00	0.	1
1.95885876D-02	0.	-2.00000000D+00	0.	1
-2.81885190D+00	-1.13464728D+02	-1.00000000D-02	7.00000000D-02	1
-2.81885190D+00	1.13464728D+02	-1.00000000D-02	-7.00000000D-02	1
-1.60650000D+01	0.	0.	0.	2
5.48250000D+00	0.	0.	0.	1

Figure 20a. DISCRET Output for Data Set 8

```
COEFFICIENTS OF NUMERATOR IN  W

(   1.000000000      )W** 6
(  -5.498289536      )W** 5
(  -1.193132632      )W** 4
(   5.691661439      )W** 3
(   .8413197480E-01  )W** 2
(  -.8386625608E-01  )W** 1
(  -.5049893189E-03  )W** 0


NUMERATOR ROOTS IN  W

(   .1194230725    ,  0.                     )
(  -.5999928005E-02,  0.                     )
(  -.1253432473    ,  0.                     )
(   1.000000000    ,  0.                     )
(  -1.017237522    ,  0.                     )
(   5.627447160    ,  0.                     )


DENOMINATOR ROOTS IN W

(   .2399539306E-01,  0.                     )
(  -.3997868031E-01,  0.                     )
(  -.2000003893E-03,   .1400000859E-02       )
(  -.2000003893E-03,  -.1400000859E-02       )
(  -1.000000000    ,  0.                     )
(  -1.000000000    ,  0.                     )


COEFFICIENTS OF DENOMINATOR IN W

(   1.000000000      )W** 6
(   2.016383288      )W** 5
(   1.031815665      )W** 4
(   .1448111464E-01  )W** 3
(  -.9516162484E-03  )W** 2
(  -.3555930123E-06  )W** 1
(  -.1918610752E-06  )W** 0


CLK=    -.6103590824E-04
CLNK=    -.6103590824E-04
CLDK=     1.000000000
```

Figure 20b. DISCRET Output for Data Set 8

```
TSAMP (SEC)=.04   DELTA (SEC)=-.004
HOLD OPTION = SLE   TRANSFORM OPTION = WP
GAIN =   -.17000000000D-01



ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--NUMERATOR

RE  -.30000000D+00   IM  0.             RE   .60000000D+01   IM  0.             RE  -.63000000D+01   IM  0.


ROOTS AFTER COMMON ROOTS HAVE BEEN DROPPED--DENOMINATOR

RE   .12000000D+01   IM  0.             RE  -.20000000D+01   IM  0.             RE  -.10000000D-01   IM   .70000000D-01
RE  -.10000000D-01   IM  -.70000000D-01 RE  0.               IM  0.             RE


PARTIAL FRACTIONS
                 N U M E R A T O R                              D E N O M I N A T O R
           REAL PART           IMAG. PART                REAL PART           IMAG. PART            EXPONENT
         1.35615214D-01       0.                       1.20000000D+00       0.                        1
         1.35885876D-02       0.                      -2.00000000D+00       0.                        1
        -2.81885190D+00      -1.13464728D+02          -1.00000000D-02       7.00000000D-02            1
        -2.81885190D+00       1.13464728D+02          -1.00000000D-02      -7.00000000D-02            1
        -1.60650000D+01       0.                       0.                   0.                        2
         5.48250000D+00       0.                       0.                   0.                        1
```

Figure 21a. DISCRET Output for Data Set 9

```
COEFFICIENTS OF NUMERATOR IN W

(     1.000000000       )W** 6
(  -274.9144768         )W** 5
( -2882.831581          )W** 4
(  711457.6799          )W** 3
(  525824.8425          )W** 2
( -26208205.03          )W** 1
( -7890458.100          )W** 0


NUMERATOR ROOTS IN W

(   5.971153627   ,  0.            )
(  -.2999964003   ,  0.            )
(  -6.267162363   ,  0.            )
(   50.00000000   ,  0.            )
(  -50.86187609   ,  0.            )
(   276.3723580   ,  0.            )


DENOMINATOR ROOTS IN W

(   1.199769653      ,  0.              )
(  -1.998934016      ,  0.              )
(  -.1000001947E-01, .7000004293E-01)
(  -.1000001947E-01, -.7000004293E-01)
(  -50.00000000      ,  0.              )
(  -50.00000000      ,  0.              )


COEFFICIENTS OF DENOMINATOR IN W

(     1.000000000      )W** 6
(   100.8191644        )W** 5
(   2579.539163        )W** 4
(   1810.139330        )W** 3
(  -5947.601553        )W** 2
(  -111.1228164        )W** 1
(  -29.97828300        )W** 0


CLK=    -.6103590024E-04
CLNK=   -.3906290128E-14
CLDK=    .6400000000E-10
```

Figure 21b. DISCRET Output for Data Set 9

SECTION V

DESCRIPTION OF TXCONV COMPUTER PROGRAM

A. INTRODUCTION

The TXCONV computer program calculates a low-rate discrete transform from a given high-rate discrete transform. The input to TXCONV is a high-rate transfer function in the z-, w-, or w′-plane and the output a low-rate transfer function in the z-, w-, or w′-plane. The general transform conversion is given by:

$$C^T(z) = \left[C^{T/m}(z_p)\right]^T \tag{76}$$

$$C^T(w) = \left[C^{T/m}(w_p)\right]^T \tag{77}$$

$$C^T(w') = \left[C^{T/m}(w_p')\right]^T \tag{78}$$

The superscript designates the sampling interval used to form the discrete transform. For example, the high-rate z_p-transform $C^{T/m}(z_p)$ is first calculated with respect to a T/m sampling period. The low-rate z-transform of this high-rate transform is then taken with respect to a sampling interval of T seconds. The result is a low-rate z-transform $C^T(z)$.

The transforms in Eqs. 76-68 are generated when the output of a system is sampled at a lower rate than the input (see Section III, Subsection D). An open-loop example of this situation is depicted in Fig. 22. The output from the physical sampler T in Fig. 22 is expressed as

$$\left[C^{T/m}\right]^T = \left[G^{T/m}R^{T/m}\right]^T \tag{79}$$

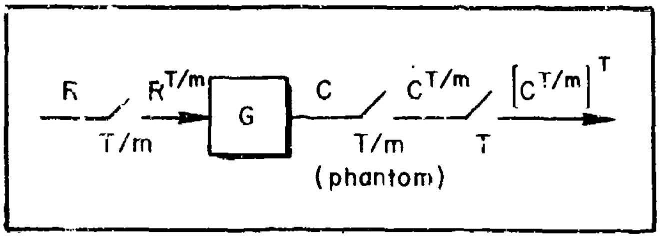


Figure 22. High-Rate Input/Low-Rate Output Sampling

The T/m phantom (fictitious) sampler in the output facilitates the formation of the $G^{T/m}R^{T/m}$ product using a common definition of the transform variable (e.g., $z_p = e^{sT/m}$). This mathematical convenience is valid since the actual output sampler T simply rejects all unwanted samples from the phantom sampler T/m. To evaluate Eq. 79, the procedure is to obtain the high-rate T/m transform of $R^{T/m}$ and multiply it by the high-rate T/m transform of $G^{T/m}$. This high-rate discrete transfer function product is the required input to the TXCONV computer program. The output from TXCONV is a low-rate discrete transfer function defined for a T sampling period.

TXCONV is written in FORTRAN for the Control Data Corporation (CDC) CYBER 175 series computer. The program can handle pole multiplicity up to three and system order up to 50th (the system order is variable and can be easily changed). There is no restriction on the number of sets of repeated poles. Double precision arithmetic is used throughout the program.

B. PROGRAM OPTIONS

TXCONV implements the conversion of a high-rate discrete transform to a low-rate discrete transform. The program accepts the zeros and poles of a high-rate discrete transfer function in the z-, w-, or w'-plane and outputs a corresponding low-rate discrete transfer function. The five available input options are described below.

1. Z Option

The program input consists of the high-rate zeros and poles in the z-plane and the low-rate (T) and high-rate (T/m) sampling periods. Prior to executing the transform conversion, the high-rate z-plane numerator and denominator polynomials are transformed to the w′-plane using the bilinear transformation $z_p = \left[(2m/T) + w_p'\right]/\left[(2m/T) - w_p'\right]$. The w′-plane denominator is then rooted to obtain the poles used in the residue calculations. In this option, all calculations are carried out in the w′-plane to minimize the numerical round-off errors. This is necessary since the poles of a z-plane function tend to migrate towards the unit circle (i.e., $z \approx 1$) as the sample rate is increased. This can introduce numerical errors in the residue computation. These inherent errors can be minimized by performing all possible calculations in the w′-plane where the poles are more reasonably separated. The resulting low-rate w′-plane transfer function is then transformed back into the z-plane using the bilinear transformation $w' = (2/T)(z - 1)/(z + 1)$. The output for this option is a low-rate discrete transfer function in the z-plane.

2. W Option

The program input is in the w-plane. All numerical calculations are carried out in the w-plane. The output is a low-rate discrete transfer function in the w-plane.

3. WP Option

The program input is in the w′-plane. All numerical calculations are carried out in the w′-plane. The output is a low-rate discrete transfer function in the w′-plane.

4. ZR Option

This option is the same as the Z option except that the w′-plane high-rate poles used in the residue calculations are obtained by direct transformation of the input z-plane poles. That is, the high-rate

w′-plane denominator is not rooted to obtain these poles as is done in the Z option. The ZR option avoids the numerical errors that may occur when rooting a polynomial.

5. ZT Option

The program input is in the z-plane. All numerical calculations are carried out in the z-plane. The output is a low-rate discrete transfer function in the z-plane. This option is limited to simple poles (i.e., pole multiplicity equal to one).

C. TRANSFORMATION EXPRESSIONS

The transformation expressions mechanized in the TXCONV computer program are given below. These expressions transform a high-rate discrete transfer function to a low-rate discrete transfer function. The mathematical derivation for these discrete transformations is presented in Section III.

1. w and w′ Plane Transformations

$$C^{T}(w) = \sum_{k} \text{residues} \left. \frac{C^{T/m}(w_p)}{A_p + w_p} \; \frac{1}{1 - X} \; \frac{2A_p}{A_p - w_p} \right|_{w_p = \text{Poles of } C^{T/m}(w_p)/(A_p + w_p)} \tag{80}$$

$$X = \left[\frac{A_p + w_p}{A_p - w_p}\right]^{m} \left[\frac{A + w}{A - w}\right]^{-1} \tag{81}$$

Alternate expressions are given by

$$C^{T}(w) = \sum_{k} \text{residues} \left. \frac{2A_p(A + w)\left[N(w_p)/D^{*}(w_p)\right]\left[1/(1 + Y^{m})\right]}{w + A[(1 - Y^{m})/(1 + Y^{m})]} \right|_{w_p = \text{Poles of } C^{T/m}(w_p)/(A_p + w_p)} \tag{82}$$

$$C^{T}(w) = \sum_{k} \text{residues } 2A_p\,(A + w) \left. \left\{ \frac{N(w_p)}{D^{*}(w_p)[(w + A) + (w - A)Y^{m}]} \right\} \right|_{w_p = \text{Poles of } C^{T/m}(w_p)/(A_p + w_p)} \tag{83}$$

where

$$C^T(w) = N(w)/D(w) \tag{84}$$

$$C^{T/m}(w_p) = N(w_p)/D(w_p) \tag{85}$$

and

$$D^*(w_p) = D(w_p)\left[(A_p + w_p)(A_p - w_p)\right] \tag{86}$$

$$Y = (A_p + w_p)/(A_p - w_p) \tag{87}$$

These transform expressions are applicable to either the w- or w′-plane. For the w-plane, the transform variables are w and w_p with $A = A_p = 1$. In the w′-plane, the transform variables become w′ and w_p' with $A = 2/T$ and $A_p = 2m/T$. In Eqs. 80-87 the following definitions apply:

m = Ratio of high-to-low sampling, (T)/(T/m)

w, w′ = Low-rate transform variables

w_p, w_p' = High-rate transform variables

$$w = \frac{z-1}{z+1}, \quad w' = \frac{2}{T}\,\frac{z-1}{z+1}, \quad z = e^{sT}$$

$$w_p = \frac{z_p-1}{z_p+1}, \quad w_p' = \frac{2}{(T/m)}\,\frac{z_p-1}{z_p+1}, \quad z_p = e^{sT/m}$$

2. <u>z-Plane Transformations</u>

$$C^T(z) = \sum_k \text{residues} \left.\frac{C^{T/m}(z_p)}{z_p}\,\frac{z}{z - z_p^m}\right|_{z_p = \text{Poles of } C^{T/m}(z_p)/z_p} \tag{88}$$

Alternate expressions are given by:

$$C^T(z) = \sum_k \text{residues} \quad z \left\{ \frac{N(z_p)}{zD^*(z_p) - z_p^m D^*(z_p)} \right\} \Bigg|_{z_p = \text{Poles of } C^{T/m}(z_p)/z_p} \tag{89}$$

$$C^T(z) = \sum_k \text{residues} \quad z \left\{ \frac{N(z_p)/D^*(z_p)}{z - z_p^m} \right\} \Bigg|_{z_p = \text{Poles of } C^{T/m}(z_p)/z_p} \tag{90}$$

where

$$C^T(z) = N(z)/D(z) \tag{91}$$

$$C^{T/m}(z_p) = N(z_p)/D(z_p) \tag{92}$$

$$D^*(z_p) = z_p D(z_p) \tag{93}$$

and

m = Ratio of high-to-low sampling, $(T)/(T/m)$

z = Low-rate transform variable ($z = e^{sT}$)

z_p = High-rate transform variable ($z_p = e^{sT/m}$)

D. GENERAL RESIDUE CALCULATION

Consider the general partial fraction expansion of a z-plane function $F(z)$ for a pole with multiplicity equal to n.

$$F(z) = \frac{N}{D} = \frac{A_n}{(z + a)^n} + \frac{A_{n-1}}{(z + a)^{n-1}} + \frac{A_{n-2}}{(z + a)^{n-2}} +$$

$$+ \frac{A_{n-3}}{(z + a)^{n-3}} + \cdots + \text{Remaining poles of } F(z) \tag{94}$$

The partial fraction expansion coefficients are given by the following equations evaluated at $z = -a$:

$$A_n = \left. \frac{(n)!N}{D(')^n} \right|_{z=-a} \tag{95}$$

$$A_{n-1} = \left. \frac{[(n+1)!/1!]N(')^1 - A_n D(')^{n+1}}{(n+1)D(')^n} \right|_{z=-a} \tag{96}$$

$$A_{n-2} = \left. \frac{[(n+2)!/2!]N(')^2 - A_n D(')^{n+2} - (n+2)A_{n-1}D(')^{n+1}}{(n+1)(n+2)D(')^n} \right|_{z=-a} \tag{97}$$

$$A_{n-3} = \left. \frac{[(n+3)!/3!]N(')^3 - A_n D(')^{n+3} - (n+3)A_{n-1}D(')^{n+2} - (n+2)(n+3)A_{n-2}D(')^{n+1}}{(n+1)(n+2)(n+3)D(')^n} \right|_{z=-a} \tag{98}$$

In Eqs. 95-98, $(')^n$ is defined as the nth derivative with respect to the transform variable z. These equations are completely general and provide the partial fraction expansion coefficients for a pole with multiplicity up to and including n = 4. The coefficient for a pole with multiplicity equal to n = 1 is given by:

$$A_1 = \left. \frac{N}{D(')^1} \right|_{z=-a} \tag{99}$$

For n = 2, the partial fraction expansion coefficients can be obtained from

$$A_1 = \left. \frac{6N(')^1 D(')^2 - 2ND(')^3}{3D(')^2 D(')^2} \right|_{z=-a} \tag{100}$$

$$A_2 = \left. \frac{2N}{D(')^2} \right|_{z=-a} \tag{101}$$

And for $n = 3$,

$$A_1 = \left. \frac{120N^{(')2}D^{(')3}D^{(')3} - 12ND^{(')3}D^{(')5} - 60N^{(')1}D^{(')3}D^{(')4} + 15N^{(')1}D^{(')4}D^{(')4}}{40D^{(')3}D^{(')3}D^{(')3}} \right|_{z=-a} \tag{102}$$

$$A_2 = \left. \frac{24N^{(')1}D^{(')3} - 6ND^{(')4}}{4D^{(')3}D^{(')3}} \right|_{z=-a} \tag{103}$$

$$A_3 = \left. \frac{6N}{D^{(')3}} \right|_{z=-a} \tag{104}$$

In Eqs. 95-104, the pole that is being evaluated $(z + a)^n$ is not explicitly factored out of the denominator polynomial D prior to taking the required derivatives or prior to the evaluation at $z = -a$. The derivation of these equations follows the procedure presented in Ref. 1. This method employs L'Hôpital's rule to eliminate the indeterminate forms that result (see Ref. 1, Appendix D for details). The general procedure is to take consecutive derivatives of the numerator and denominator polynomials until a determinate form is obtained. For example, in the $n = 1$ case, we have

$$F(z) = \frac{N}{D} = \frac{A_1}{(z + a)} + \cdots + \text{Remaining poles of } F(z) \tag{105}$$

Then

$$A_1 = \left. \frac{(z + a)N}{D} \right|_{z=-a} \tag{106}$$

If Eq. 106 is evaluated at $z = -a$ without first explicitly factoring out the $(z + a)$ factor in the denominator and cancelling it with the numerator factor $(z + a)$, an indeterminate form (0/0) will result. Applying L'Hôpital's rule (i.e., taking separate derivatives of the numerator and denominator with respect to z) gives

$$A_1 = \left.\frac{(z + a)N' + N}{D'}\right|_{z=-a}$$

$$= \left.\frac{N}{D'}\right|_{z=-a} \tag{107}$$

Notice that for any pole with a multiplicity of n = 1, the evaluation of the first derivative of the denominator produces a finite result. That is,

$$D'\big|_{z=-a} \neq 0.0 \tag{108}$$

This can be stated in more general terms for any pole with multiplicity equal to n as

$$D(')^k\big|_{z=-a} \neq 0.0 \quad , \quad (z + a)^n \quad , \quad k \geq n \tag{109}$$

and

$$D(')^k\big|_{z=-a} = 0.0 \quad , \quad (z + a)^n \quad , \quad k < n \tag{110}$$

where, again, $(')^k$ is defined as the kth derivative of the denominator D.

For example, if

$$F(z) = \frac{N}{D} = \frac{(z + 2)(z + 3)}{(z + 5)(z + 10)} = \frac{A_1}{(z + 5)} + \frac{B_1}{(z + 10)} \tag{111}$$

Then

$$D = z^2 + 15z + 50 \tag{112}$$

$$D' = 2z + 15 \tag{113}$$

and

$$A_1 = \left.\frac{N}{D'}\right|_{z=-10} = \left.\frac{z^2 + 5z + 6}{2z + 15}\right|_{z=-10} = -13.2 \tag{114}$$

$$B_1 = \left.\frac{N}{D'}\right|_{z=-5} = \left.\frac{z^2 + 5z + 6}{2z - 15}\right|_{z=-5} = 1.2 \tag{115}$$

The evaluation of the residues in the TXCONV computer program is accomplished using Eqs. 99, 100, and 102. It is recognized that the residue for a pole with multiplicity equal to n is given by the partial fraction expansion coetticient associated with the $(z + a)$ term. Applying Eqs. 99, 100, and 102 to Eqs. 80 and 88 results in closed-form solutions which are functions of the separate derivatives of the numerator and denominator polynomials for the given high-rate discrete transfer function. The actual expressions mechanized in TXCONV are developed in the next subsection.

E. MECHANIZATION SCHEME

The transformation expression in the z-plane is only mechanized for poles with multiplicity equal to one (ZT option). This transformation is coded in subroutine ZMULT1. As explained previously, for pole multiplicity greater than one $(n > 1)$, the input high-rate z-plane transfer function is first transformed to the w′-plane prior to performing the numerical calculations (Z and ZR options). This procedure improves the accuracy of the results by reducing to a minimum the errors introduced by numerical round-off. The z-plane convesion for option ZT is formed by applying Eq. 99 to Eq. 88. The resul is Eq. 116 (see Subsections C and D for definition of terms).

$$C^T(z) = \sum_k \text{residues } z \left\{ \frac{N(z_p)/D^*(z_p)'}{z - z_p^m} \right\} \Bigg|_{z_p = \text{Poles of } D^*(z_p)} \tag{116}$$

where

$$D^*(z_p) = z_p D(z_p) \tag{117}$$

$$m = (T)/(T/m) \tag{118}$$

$$z = e^{sT} \quad , \quad z_p = e^{sT/m} \tag{119}$$

For all options except ZT (i.e., Z, W, WP, and ZR), separate mechanization expressions are used for poles with multiplicity equal to one, two, and three. These expressions are coded in the subroutines WMULT1, WMULT2, and WMULT3 for n = 1, 2, and 3, respectively. Equations 99, 100, and 102 are individually applied to Eq. 80 to form these expressions. The resulting transform conversion equations (Eqs. 120, 121, and 124) are applicable to either the w- or w′-plane. The definition of terms for w or w′ implementation is given in Subsections C and D. The specific expressions that are mechanized are outlined below.

Multiplicity n = 1

$$C^T(w) = \sum_k \text{residues } 2A_p(A+W) \left\{ \frac{N(w_p)/D^*(w_p)'[1/(1+Y^m)]}{w + \text{POLE}} \right\} \Bigg|_{w_p = \text{Poles of } D(w_p)(A_p + w_p)} \tag{120}$$

Multiplicity of n = 2

$$C^T(w) = \sum_k \text{residues } 2A_p(A+W)\,\frac{(N1)w + (N2)}{w + (POLE)^2} \Bigg|_{w_p = \text{Poles of } D(w_p)(A_p+w_p)} \qquad (121)$$

where

$$N1 = \frac{2N'(w_p)}{D^{*''}(w_p)(1+Y^m)} - \frac{2N(w_p)D^{*'''}(w_p)}{3(D^{*''}(w_p))^2(1+Y^m)} - \frac{2mN(w_p)Y^{m-1}Y'}{D^{*''}(w_p)(1+Y^m)^2}, \qquad (122)$$

$$N2 = \frac{2N'(w_p)(POLE)}{D^{*''}(w_p)(1+Y^m)} - \frac{2N(w_p)D^{*'''}(w_p)(POLE)}{3(D^{*''}(w_p))^2(1+Y^m)} + \frac{2mAN(w_p)Y^{m-1}Y'}{D^{*''}(w_p)(1+Y^m)^2} \qquad (123)$$

Multiplicity of n = 3

$$C^T(w) = \sum_k \text{residues } 2A_p(A+w) \left\{ \frac{\begin{array}{l}(N1+N2)[w^2 + 2(POLE)w + (POLE)^2] \\ \quad + (N3+N4)[w^2 + (POLE - A)w - A(POLE)] \\ \quad + (N5)[w^2 - (2A)w + A^2]\end{array}}{w + (POLE)^3} \right\} \Bigg|_{\substack{w_p = \text{Poles of} \\ D(w_p)(A_p+w_p)}} \qquad (124)$$

where

$$N1 = \frac{3N''(w_p)}{D^{*'''}(w_p)(1+Y^m)} - \frac{1.5N'(w_p)D^{*''''}(w_p)}{(D^{*'''}(w_p))^2(1+Y^m)} \qquad (125)$$

$$N2 = \frac{.375N(w_p)(D^{*''''}(w_p))^2}{(D^{*'''}(w_p))^3(1+Y^m)} - \frac{0.3N(w_p)D^{*'''''}(w_p)}{(D^{*'''}(w_p))^2(1+Y^m)} \qquad (126)$$

$$N3 = \frac{3mN(w_p)Y^{m-1}Y'D^{*''''}}{(D^{*'''}(w_p))^2(1+Y^m)^2} - \frac{6mN'(w_p)Y^{m-1}Y'}{D^{*'''}(w_p)(1+Y^m)^2} - \frac{1.5mN(w_p)Y^{m-1}Y'D^{*''''}(w_p)}{(D^{*'''}(w_p))^2(1+Y^m)^2} \qquad (127)$$

$$N4 = -\frac{3m(m-1)N(w_p)Y^{m-2}(Y')^2}{D^{*'''}(w_p)(1+Y^m)^2} - \frac{3mN(w_p)Y^{m-1}Y''}{D^{*'''}(w_p)(1+Y^m)^2} \qquad (128)$$

$$N5 = \frac{6m^2N(w_p)(Y^{m-1})^2(Y')^2}{D^{*'''}(w_p)(1+Y^m)^3} \qquad (129)$$

For Eqs. 120-129, the following relationships and definitions apply:

$$\text{POLE} = A\left[(1 - Y^m)/(1 + Y^m)\right] \tag{130}$$

$$Y = (A_p + w_p)/(A_p - w_p) \tag{131}$$

$$Y' = 2A_p/(A_p - w_p)^2 \tag{132}$$

$$Y'' = 4A_p/(A_p - w_p)^3 \tag{133}$$

$$D^*(w_p) = D(w_p)(A_p + w_p)(A_p - w_p) \tag{134}$$

$$m = (T)/(T/m) \tag{135}$$

F. ROOT CONVERSION BETWEEN s, z, w, or w′ PLANES

The conversion of the poles between the s-, z-, w-, or w′-plane is mechanized in the subroutine SZWROOT. The algebraic relationships implemented in SZWROOT allow direct transformation of the poles from one complex plane to another. Direct conversion of the zeros on a one-for-one basis is not normally possible. One exception is the conversion between the w- and w′-plane where both the zeros and poles transform directly ($w' = 2w/T$). It is also possible to directly transform the zeros and poles from the w- or w′-plane to the z-plane. However, the inverse of this is not normally true (i.e., z-plane to w- or w′-plane direct conversion of zeros).

The complete conversion between any of the complex planes noted above can be implemented by transforming the complex function instead of its zeros and poles. Although, in some cases, physically unrealizable functions may result. The general procedure is to transform the numerator and denominator polynomials (or partial fraction expansion terms) and then factor the results to obtain the transformed zeros and poles. This procedure is normally less desirable than direct conversion of the zeros and poles, since the particular mathematical calculations (e.g.,

numerical factoring; multiplication and addition of polynomials) may introduce additional numerical errors.

The subroutine SZWROOT is used in the TXCONV computer program to transform the poles between the z- and w'-planes. The other conversions available in SZWROOT are not utilized. Conversion of the poles between the z- and w'-planes occur only in the ZR option. The algebraic conversion relationships implemented in SZWROOT are outlined below. The following definitions apply to the root conversion equations:

$$z = e^{sT} = \frac{1 + w}{1 - w} = \frac{(2/T) + w'}{(2/T) - w'} \tag{136}$$

$$w = \frac{z - 1}{z + 1} = (T/2)w' \tag{137}$$

$$w' = \frac{2}{T} \frac{z - 1}{z + 1} = (2/T)w \tag{138}$$

$$s = X_s + jYs \quad , \quad z = X_z + jY_z \tag{139}$$

$$w = X_w + jY_w \quad , \quad w' = X_{w'} + jY_{w'} \tag{140}$$

<u>s-Plane to z-Plane</u>

$$X_z = e^{X_s T} \cos Y_s T \tag{141}$$

$$Y_z = e^{X_s T} \sin Y_s T \tag{142}$$

w-Plane to z-Plane

$$X_z = \frac{1 - X_w^2 - Y_w^2}{(1 - X_w)^2 + Y_w^2} \tag{143}$$

$$Y_z = \frac{2Y_w}{(1 - X_w)^2 + Y_w^2} \tag{144}$$

w′-Plane to z-Plane

$$X_z = \frac{1 - [(T/2)X_{w'}]^2 - [(T/2)Y_{w'}]^2}{[1 - (T/2)X_{w'}]^2 + [(T/2)Y_{w'}]^2} \tag{145}$$

$$Y_z = \frac{2[(T/2)Y_{w'}]}{[1 - (T/2)X_{w'}]^2 + [(T/2)Y_{w'}]^2} \tag{146}$$

z-Plane to s-Plane

$$X_s = \frac{1}{2T} \ln \left(X_z^2 + Y_z^2\right) \tag{147}$$

$$Y_s = \frac{1}{T} \tan^{-1} (Y_z/X_z) \tag{148}$$

w-Plane to s-Plane

$$X_s = \frac{1}{2T} \ln \frac{(1 + X_w)^2 + Y_w^2}{(1 - X_w)^2 + Y_w^2} \tag{149}$$

$$Y_s = \frac{1}{T} \tan^{-1} \frac{2Y_w}{1 - X_w^2 - Y_w^2} \tag{150}$$

w′-Plane to s-Plane

$$X_s = \frac{1}{2T} \ln \frac{[1 + (T/2)X_{w'}]^2 + [(T/2)Y_{w'}]^2}{[1 - (T/2)X_{w'}]^2 + [(T/2)Y_{w'}]^2} \tag{151}$$

$$Y_s = \frac{1}{T} \tan^{-1} \frac{2[(T/2)Y_{w'}]}{1 - [(T/2)X_{w'}]^2 - [(T/2)Y_{w'}]^2} \tag{152}$$

s-Plane to w-Plane

$$X_w = \frac{e^{X_s T} - e^{-X_s T}}{e^{X_s T} + e^{-X_s T} + 2 \cos Y_s T} \tag{153}$$

$$Y_w = \frac{2 \sin Y_s T}{e^{X_s T} + e^{-X_s T} + 2 \cos Y_s T} \tag{154}$$

z-Plane to w-Plane

$$X_w = \frac{X_z^2 + Y_z^2 - 1}{(X_z + 1)^2 + Y_z^2} \tag{155}$$

$$Y_w = \frac{2Y_z}{(X_z + 1)^2 + Y_z^2} \tag{156}$$

w′-Plane to w-Plane

$$X_w = \frac{T}{2} X_{w'} \tag{157}$$

$$Y_w = \frac{T}{2} Y_{w'} \tag{158}$$

s-Plane to w′-Plane

$$X_{w'} = \frac{2}{T} \frac{e^{X_s T} - e^{-X_s T}}{e^{X_s T} + e^{-X_s T} + 2 \cos Y_s T} \tag{159}$$

$$Y_{w'} = \frac{2}{T} \frac{2 \sin Y_s T}{e^{X_s T} + e^{-X_s T} + 2 \cos Y_s T} \tag{160}$$

z-Plane to w′-Plane

$$X_{w'} = \frac{2}{T} \frac{X_z^2 + Y_z^2 - 1}{(X_z + 1)^2 + Y_z^2} \tag{161}$$

$$Y_{w'} = \frac{2}{T} \frac{2Y_z}{(X_z + 1)^2 + Y_z^2} \tag{162}$$

w-Plane to w′-Plane

$$X_{w'} = \frac{2}{T} w \tag{163}$$

$$Y_{w'} = \frac{2}{T} w \tag{164}$$

G. PROGRAM STRUCTURE

The basic structure of the TXCONV computer program contains a main program TXCONV and seven primary subroutines ZMULT1, WMULT1, WMULT2, WMULT3, RES1, RES2, and RES3. The program modules along with the 20 supporting subroutines are depicted in Fig. 23. The main program TXCONV first calls ZMULT1, WMULT1, WMULT2, and WMULT3 to evaluate the individual residues of the high-rate poles. Subroutines RES1, RES2, and RES3 are then called to combine these residues to form the low-rate discrete transfer function. ZMULT1 calculates the residues for poles with multiplicity equal to one ($n = 1$). This subroutine mechanizes Eq. 116 in the z-plane. WMULT1, WMULT2, and WMULT3 calculate the residues for poles with multiplicity equal to one, two, and three ($n = 1, 2, 3$), respectively. These subroutines mechanize Eqs. 120, 121, and 124 in the w- or w′-plane.

The TXCONV computer program can be operated in a standard program-subroutine structure (i.e., a main program followed by its subroutine) or in an overlay structure. The source listing for TXCONV in Volume III is set up in a program-subroutine structure. However, this listing also includes (in the comment code) the required changes to run the program in an overlay structure. Dividing the program into overlays reduces the amount of computer memory required to execute the program. The overlay code is highlighted with a star (*) character in column one (which makes the code inactive comment). Removing this code from comment will allow overlay operation. To complete this turnover, the TXCONV program card and subroutine cards for ZMULT1, WMULT1, WMULT2, WMULT3, RES1, RES2, and RES3 must be deleted. In addition, the call statements for these subroutines (which are now overlays) must also be deleted. These call statements are located in the "Master Do Loop for Residues" section in the main program. The overlay code creates a main overlay, a primary overlay, and seven secondary overlays. This overlay structure is outlined below:

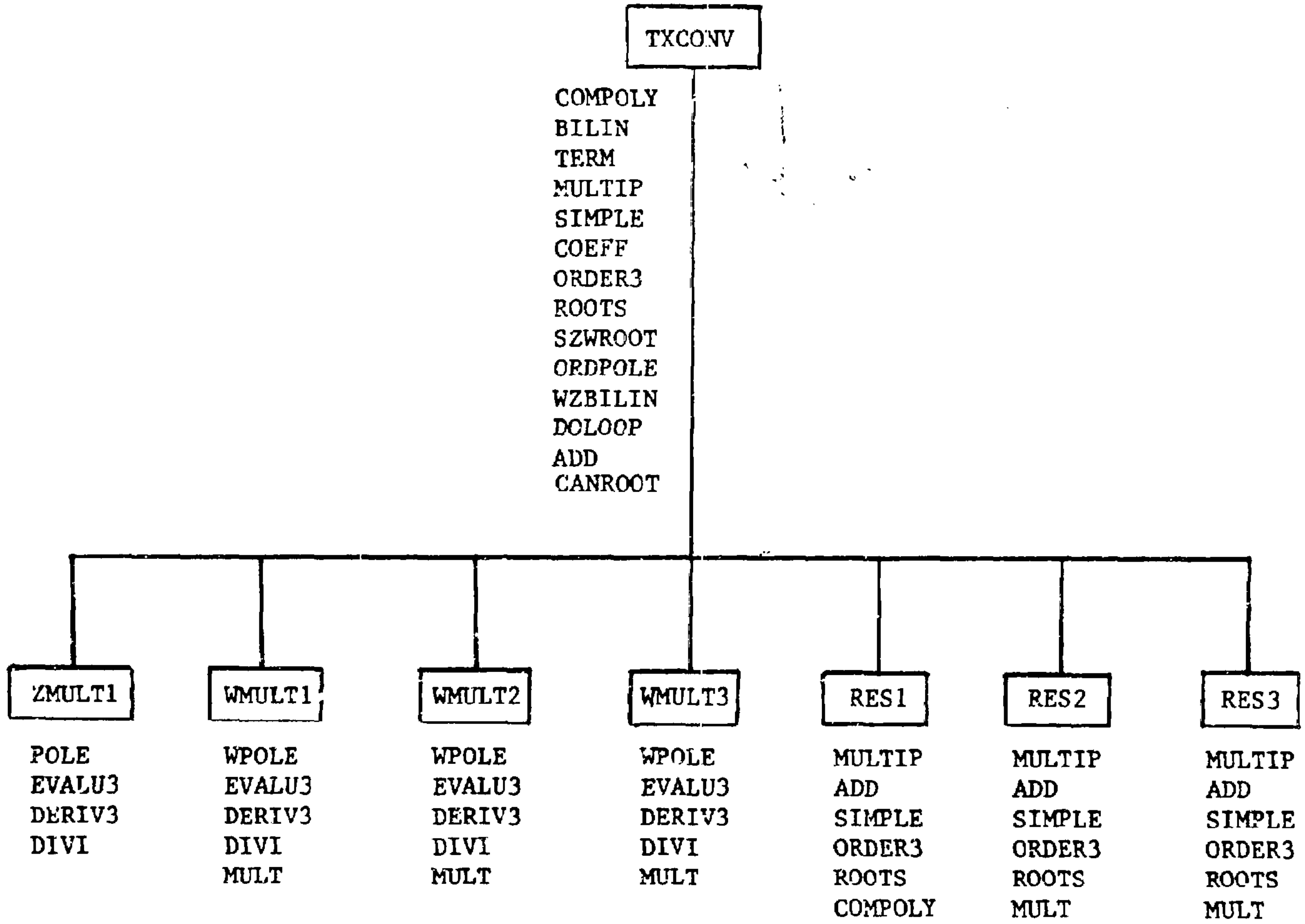


Figure 23. TXCONV Program Structure

Overlay	Program Name
(TXCONV,0,0,0)	TXCONV
(27,0)	MAIN
(27,1)	ZMULT1
(27,2)	WMULT1
(27,3)	WMULT2
(27,4)	WMULT3
(27,5)	RES1
(27,6)	RES2
(27,7)	RES3

Parameters are passed between the main program and the seven primary subroutines entirely in a common data structure. This lack of formal parameter passing via subroutine arguments was initiated to allow the program to be easily converted to an overlay structure in the TOTAL (Ref. 13) computer program at AFWAL/FIGC. This permits interactive operation of TXCONV as an option in TOTAL.

H. DESCRIPTION OF SUBROUTINES

This subsection contains a brief definition of the routines used in the TXCONV computer program. A detailed description of each routine is documented in the COMMENT section of the source code. This comment code defines all subroutine arguments and the major internal variables in each routine. In addition, descriptive comments are included throughout each routine. The present source code (Volume III) is set up to handle 50th order systems. The comment code in the main program TXCONV lists the required changes to the program DIMENSION statements to alter the maximum system order allowed. The purpose of each variable and array in labeled COMMON is also included in the source code for the main program.

Program TXCONV. This is the main program for the TXCONV computer program. It reads the input from data cards and transfers the data to

internal variables and arrays that are located in the COMMON data structure. It adds appropriate denominator poles ($z = 0$, $w = -1$, or $w' = 2m/T$) to the residue expressions. The call statements for the seven primary subroutines (ZMULT1, WMULT1, WMULT2, WMULT3, RES1, RES2, and RES3) are located in the main program. The final formation of the output low-rate discrete transfer function is accomplished in this program module. The individual low-rate discrete transfer functions from subroutines RES1, RES2, and RES3 are combined to form the final output.

Subroutine ZMULT1. This subroutine calculates the residues for poles with multiplicity equal to one ($n = 1$). ZMULT1 mechanizes the z-plane residue expression in Eq. 116.

Subroutine WMULT1. This subroutine calculates the residues for poles with multiplicity equal to one ($n = 1$). WMULT1 mechanizes the w-plane/w′-plane residue expression in Eq. 120.

Subroutine WMULT2. This subroutine calculates the residues for poles with multiplicity equal to two ($n = 2$). WMULT2 mechanizes the w-plane/w′-plane residue expression in Eq. 121.

Subroutine WMULT3. This subroutine calculates the residues for poles with multiplicity equal to three ($n = 3$). WMULT3 mechanizes the w-plane/w′-plane residue expression in Eq. 124.

Subroutine RES1. This subroutine forms the overall low-rate discrete transfer function for 1st order poles. The $n = 1$ (multiplicity equal one) residues that are calculated in subroutines ZMULT1 or WMULT1 are combined to form this transfer function.

Subroutine RES2. This subroutine forms the overall low-rate discrete transfer function for 2nd order poles. The $n = 2$ (multiplicity equal two) residues that are calculated in subroutine WMULT2 are combined to form this transfer function.

Subroutine RES3. This subroutine forms the overall low-rate discrete transfer function for 3rd order poles. The $n = 3$ (multiplicity equal three) residues that are calculated in subroutine WMULT3 are combined to form this transfer function.

Subroutine COMPOLY. This subroutine forms a polynomial from a set of roots. Both real and imaginary coefficients are calculated.

Subroutine BILIN. This subroutine performs the general bilinear transformation from one complex plane to another.

Subroutine TERM. This subroutine calculates the individual terms used in the bilinear transformation mechanized in subroutine BILIN.

Subroutine WZBILIN. This subroutine initiates the specific bilinear transformation from the w′-plane to the z-plane. The actual transformation is carried out in subroutine BILIN.

Subroutine MULTIP. This subroutine multiplies two polynomials.

Subroutine ADD. This subroutine adds two polynomials.

Subroutine SIMPLE. This subroutine simplifies a polynomial by adding coefficients of like powers.

Subroutine COEFF. This subroutine adds the missing power terms in a polynomial by inserting a zero coefficient with the appropriate power and moving the original terms to make room for the missing terms.

Subroutine ORDER3. This subroutine orders a polynomial in descending powers.

Subroutine ROOTS. This subroutine finds the roots of a polynomial with complex coefficients.

Subroutine SZWROOT. This subroutine performs the root conversion between the s-, z-, w-, and w′-complex planes.

Subroutine ORDPOLE. This subroutine checks for multiple poles and stores the multiplicity in an array. The extra multiple poles are deleted and only a single copy of each pole is stored in the output array.

Subroutine POLE. This subroutine calculates a low-rate pole in the z-plane from a given high-rate pole in the z-plane.

Subroutine WPOLE. This subroutine calculates a low-rate pole in the w- or w′-plane from a given high-rate pole in the w- or w′-plane.

Subroutine DERIV3. This subroutine takes the derivative of a polynomial.

Subroutine EVALU3. This subroutine evaluates a polynomial for a given root.

Subroutine DIVI. This subroutines divides two complex numbers.

Subroutine MULT. This subroutine multiplies two complex numbers.

Subroutine DOLOOP. This subroutine implements a standard DO LOOP to transfer one array into another array.

Subroutine CANROOT. This subroutine cancels equal zeros and poles according to a specified tolerance. Separate tolerances are provided for the real and imaginary parts of each root.

I. PROGRAM OPERATING INSTRUCTIONS

This subsection presents the input and output data structure for the TXCONV computer program. Most input data are in free format with separators rather than in fixed-size fields. The one exception is the alphanumeric input which selects the transformation option (Z, W, WP, ZR, or ZT). The free-format input data consist of a string of values separated by one or more blanks, or by a comma or slash, either of which may be preceded or followed by any number of blanks. A line boundary, such as an end of record or end of card, also serves as a value separator (Ref. 14).

The first section of input data contains the parameters that define the high-rate discrete transfer function, the high-rate sampling period (TIN), and the low-rate sampling period (TOUT). These data are placed on the first two data cards in a free format. The alphanumeric code for the transformation option is placed on data card three in an A2 format. (See Subsection B for an explanation of the transformation options.) The remaining data are again free format and consist of the zeros and poles of the high-rate transfer function. The order of this transfer function (i.e., order of the denominator polynomial) can be equal to or less than 50 with pole multiplicity up to and including three. The

zeros and poles are input sequentially in a free format (starting on data card four) on as many data cards as necessary. The real and imaginary parts are separated with a valid separator (i.e., a comma, a slash, or one or more blanks). The zeros are inserted first followed by the poles. For real roots, 0.0 must be input for the imaginary part. The required input data are outlined below:

Data Card	Data Items	Format
1	GAIN, NZEROS, NPOLES	Free format
2	TIN, TOUT	Free format
3	Transformation option	A2
4	Zeros(real,imag)	Free format
nth	Poles(real,imag)	Free format

The following definitions apply to the data items listed above:

GAIN - High-rate transfer function gain

NZEROS - Number of zeros

NPOLES - Number of poles

TIN - High-rate sampling period (sec)

TOUT - Low-rate sampling period (sec)

Transformation option - Z, W, WP, ZR, or ZT

Zeros(real,imag) - Real and imaginary parts of high-rate zeros

Poles(real,imag) - Real and imaginary parts of high-rate poles

The output data from TXCONV are divided into two main sections. The first section contains a listing of the input variables associated with the high-rate discrete transfer function. These variables include the ratio of the sampling periods (RATIO); the high-rate (TIN) and low-rate

(TOUT) sampling periods; the high-rate transfer function gain (GAIN); and the number of input zeros (NZEROS) and poles (NPOLES). The program also prints the numerator and denominator polynomials and roots for the high-rate discrete transfer function. A listing of the high-rate poles with their multiplicity is then printed. An additional high-rate pole (required by the transformation process) at z = 0.0, w = 1.0, or w′ = 2/TIN is included in this listing (see Section III). The second section of output data deals with the low-rate discrete transfer function. The numerator and denominator polynomials and their roots are printed. The low-rate transfer function gain and a second listing of the high-rate and low-rate sampling periods are also given.

The fast-input/slow-output sampled system in Fig. 24 will be used to illustrate the input and output data structure for TXCONV. The procedure for this example is typical for closed-loop systems employing fast-input/slow-output sampling.

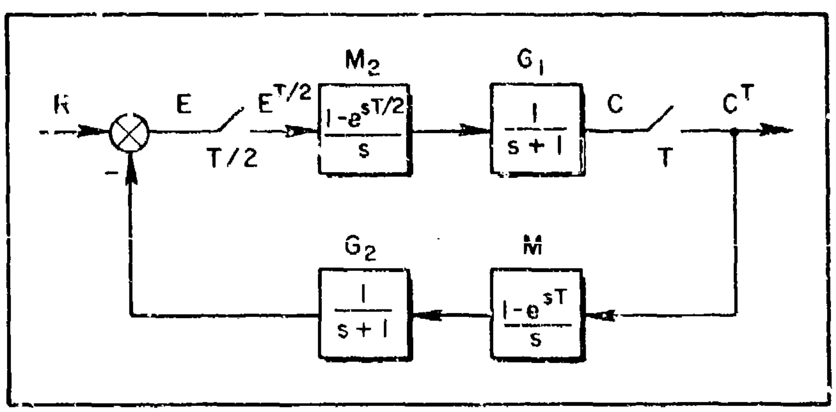


Figure 24. Fast-Input/Slow-Output Sampled System

The output equation for Fig. 24 is given by

$$C = (G_1M_2)E^{T/2} \tag{165}$$

where

$$E^{T/2} = R^{T/2} - (G_2M)^{T/2}(G_1M_2E^{T/2})^T \tag{166}$$

To solve for the $E^{T/2}$ signal in Eq. 166, premultiply by G_1M_2 and take the T transform of both sides of the resulting equation (or sample both sides of the equation at a T interval). The result is

$$(G_1M_2E^{T/2})^T = (G_1M_2R^{T/2})^T - [(G_1M_2)(G_2M)^{T/2}]^T(G_1M_2E^{T/2})^T \tag{167}$$

Rearranging Eq. 167,

$$(G_1M_2E^{T/2})^T = \left\{ I + [(G_1M_2)(G_2M)^{T/2}]^T \right\}^{-1} (G_1M_2R^{T/2})^T \tag{168}$$

and substituting Eq. 168 into Eq. 166 produces Eq. 169.

$$E^{T/2} = R^{T/2} - (G_2M)^{T/2} \left\{ I + [(G_1M_2)(G_2M)^{T/2}]^T \right\}^{-1} (G_1M_2R^{T/2})^T \tag{169}$$

Finally, substituting Eq. 169 into Eq. 165 gives the output equations for Fig. 24.

$$C = (G_1M_2)R^{T/2} - (G_1M_2)^{T/2}(G_2M)^{T/2} \left\{ I + [(G_1M_2)(G_2M)^{T/2}]^T \right\}^{-1} (G_1M_2R^{T/2})^T \tag{170}$$

To illustrate the operation of the TXCONV computer program, we calculate the term

$$[(G_1M_2)(G_2M)^{T/2}]^T \tag{171}$$

Let $z = e^{sT/2}$, and introduce a phantom T/2 sampler to Eq. 171. This mathematical operation is depicted in Fig. 25. This step is valid since the T output sampler simply rejects all the unwanted samples from the T/2 sampler. Equation 171 then becomes

$$[(G_1M_2)^{T/2}(G_2M)^{T/2}]^T \tag{172}$$

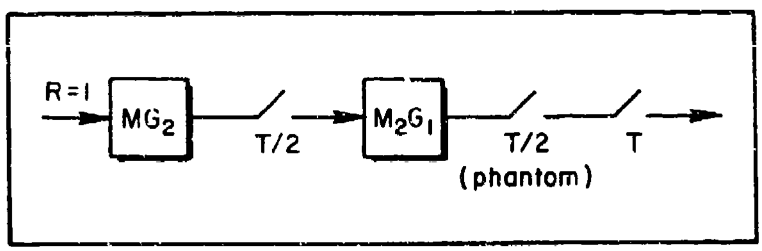


Figure 25. Phantom Sampler Concept

We next choose a low-rate sampling period of $T = -\ln(0.81)$. This gives a set of convenient numbers when Eq. 172 is evaluated. Inserting the transforms depicted in Fig. 24 into Eq. 172 produces

$$(G_1M_2)^{T/2} = \frac{1 - e^{-T/2}}{z - e^{-T/2}} \quad , \quad (G_2M)^{T/2} = \frac{1 - e^{-T/2}}{z - e^{-T/2}} \; \frac{z + 1}{z} \tag{173}$$

and

$$[(G_1M_2)^{T/2}(G_2M)^{T/2}]^T = \left[\frac{(1 - e^{-T/2})^2(z + 1)}{z(z - e^{-T/2})^2}\right]^T , \quad z = e^{sT/2}$$

$$= \left[\frac{0.01(z + 1)}{z(z - 0.9)^2}\right]^T \tag{174}$$

Equation 174 can be solved by calculating the residues of the following expression (see Eqs. 36 and 88).

$$\frac{1}{2\pi j} \int_\Gamma \frac{(1 - e^{-T/2})^2(z_p + 1)}{z_p(z_p - e^{-T/2})^2} \; \frac{z}{(z - z_p^2)} \; \frac{dz_p}{z_p} \tag{175}$$

The residues for the double poles at $z_p = 0.0$ and $z_p = e^{-T/2}$ are:

$$\mathrm{Res}\Big|_{z_p=0.0} = \frac{d}{dz_p}\left[\frac{(z_p + 1)z}{(z_p - e^{-T/2})^2(z - z_p^2)}\right]\Bigg|_{z_p=0.0} = \frac{e^{-T} + 2e^{-T/2}}{e^{-2T}} \tag{176}$$

$$\mathrm{Res}\Big|_{z_p=e^{-T/2}} = \frac{d}{dz_p}\left[\frac{(z_p + 1)z}{(z_p)^2(z - z_p^2)}\right]\Bigg|_{z_p=e^{-T/2}} = \frac{-(e^{-T} + 2e^{-T/2})z^2 + (3e^{-2T} + 4e^{-3T/2})z}{e^{-2T}(z - e^{-T})^2} \tag{177}$$

Combining Eqs. 176 and 177 gives the low-rate discrete transfer function in Eq. 178.

$$(1 - e^{-T/2})^2 \left[\mathrm{Res}\Big|_{z_p = 0.0} + \mathrm{Res}\Big|_{z_p = e^{-T/2}} \right] = \frac{(1 - e^{-T/2})^2 \; z + (e^{-T} + 2e^{-T/2})}{(z - e^{-T})^2} \quad , \quad z = e^{sT}$$

$$= \frac{0.01(z + 2.61)}{(z - 0.81)^2} \qquad (178)$$

Equation 174 is now solved using the TXCONV computer program. The required input data are outlined below:

Data Card	Data Items
1	.01,1,3
2	.1053605155,.210721031
3	ZR
4	-1.0,0.0,0.0,0.0
5	.9,0.0,.9,0.0

The output for this example is shown in Fig. 26. Both the high-rate and low-rate z-plane transfer functions are printed.

A second example will consider the w′-plane high-rate to low-rate transform conversion in Eq. 179.

$$\left[KG(w')^{T/3}\right]^T \qquad (179)$$

where

$$K = .01274481960$$

$$T = .04 \text{ (sec)}$$

$$T/3 = .12 \text{ (sec)}$$

The high-rate zeros and poles for the w′-plane transfer function $G(w')^{T/3}$ are given by:

Zeros (real,imag)	Poles (real,imag)
(-.9999481996 , 0.0)	(-.5087865665 , .3042290447)
(.001830897924, 0.0)	(-.5087865665 , -.3042290447)
(-.5657507592 , 6.861565379)	(-.001726986844, 0.0)
(-.5657507592 , -6.861565379)	(-2.169528987 , 3.364297796)
(-4.956749255 , 0.0)	(-2.169528987 , -3.364297796)
(-9.892804162 , 0.0)	(-12.82548298 , 0.0)
(-14.00440188 , 0.0)	(-16.86376037 , -12.01331830)
(-15.45466095 , 0.0)	(-16.86376037 , 12.01331830)
(50.000000000 , 0.0)	(-10.52412126 , 0.0)

A listing of the input data for this example is outlined below:

Data Cards	
1	.01274481960,9,9
2	.04,.12
3	WP
4	Zeros (real,imag)
nth	Poles (real,imag)

The output for this w′-plane example is shown in Fig. 27. Figure 27a contains the high-rate w′-plane transfer function input and Fig. 27b the low-rate w′-plane transfer function output.

The high-rate w′ transfer function $KG(w')^{T/3}$ was obtained from the DISCRET computer program (Section IV) using the sampled continuous system defined in Eq. 180.

$$KG(w')^{T/3} = \left[K_1 G_1(s) M_3(s)\right]^{T/3} \qquad (180)$$

In Eq. 180,

$$M_3(s) = \frac{1 - e^{-sT/3}}{s} \quad , \qquad K_1 = -.6483736462 \;,$$

and the s-plane zeros and poles for $G_1(s)$ are given by:

Zeros (real,imag)	Poles (real,imag)
(-1.000000000 , 0.0)	(-.5087852889 , .3042567928)
(.001830897352, 0.0)	(-.5087852889 , -.3042567928)
(-5.000000000 , 0.0)	(-.001726986844,0.0)
(-.5008733927 , 6.832938756)	(-2.161077353 , 3.365523190)
(-.5008733927 , -6.832938756)	(-2.161077353 , -3.365523190)
(-15.00000000 , 0.0)	(-13.11843178 , 0.0)
(-15.00000000 , 0.0)	(-16.40426112 , -13.13942724)
(-10.000000000 , 0.0)	(-16.40426112 , 13.13942724)
	(-10.68380409 , 0.0)

These parameters were used with the WP option in DISCRET to obtain the w′-plane transfer function $KG(w')^{T/3}$.

```
RATIO =  2.0000000          TIN =  .10536052          TOUT =  .21072103
GAIN =  .10000000E-01       NZEROS =  1               NPOLES =  3

        ZR-PLANE  HIGH-RATE NUMERATOR                          ZR-PLANE  LOW-RATE NUMERATOR

    (  1.000000000      )ZR**  1                       (  1.000000000      )ZR**  1
    (  1.000000000      )ZR**  0                       (  2.610000000      )ZR**  0

        ZR-PLANE  HIGH-RATE DENOMINATOR                        ZR-PLANE  LOW-RATE DENOMINATOR

    (  1.000000000      )ZR**  3                       (  1.000000000      )ZR**  2
    ( -1.800000000      )ZR**  2                       ( -1.620000000      )ZR**  1
    (   .8100000000     )ZR**  1                       (   .6561000000     )ZR**  0
    (  0.               )ZR**  0

          ZR-PLANE  HIGH-RATE ZEROS                            ZR-PLANE  LOW-RATE ZEROS

 (  -1.000000000    ,  0.                )           (  -2.610000000    ,  0.                )

          ZR-PLANE  HIGH-RATE POLES                            ZR-PLANE  LOW-RATE POLES

 (  0.               ,  0.                )           (   .8100000000    ,  0.                )
 (   .9000000000     ,  0.                )           (   .8100000000    ,  0.                )
 (   .9000000000     ,  0.                )           LOW-RATE ZR-PLANE  GAIN =  .1000000000E-01
                                                      HIGH-RATE SAMPLING PERIOD (SEC) =  .1053605156
          WP-PLANE  HIGH-RATE POLES                   LOW-RATE  SAMPLING PERIOD (SEC) =  .2107210310

 (  -18.98244318     ,  0.                )
 (  -.9990759574     ,  0.                )
              MULTIPLICITY =  2
              MULTIPLICITY =  2
```

Figure 26. TYCONV z-Plane Output for Example 1

```
RATIO = 3.0000000        TIN = .40000000E-01    TOUT = .12000000
GAIN = .12744820E-01     NZEROS = 9             NPOLES = 9

       UP-PLANE  HIGH-RATE NUMERATOR

       (    1.000000000        )UP**   9
       (   -3.561764933        )UP**   8
       (   -1476.099237        )UP**   7
       (   -33937.62129        )UP**   6
       (   -360908.2506        )UP**   5
       (   -2555174.944        )UP**   4
       (   -13360823.74        )UP**   3
       (   -36266773.15        )UP**   2
       (   -25085803.51        )UP**   1
       (    46051.20061        )UP**   0

       UP-PLANE  HIGH-RATE DENOMINATOR

       (    1.000000000        )UP**   9
       (    62.43548387        )UP**   8
       (    1677.848439        )UP**   7
       (    23007.95963        )UP**   6
       (    165029.2738        )UP**   5
       (    637450.6526        )UP**   4
       (    1471529.836        )UP**   3
       (    1116395.305        )UP**   2
       (    327799.6203        )UP**   1
       (    562.7835748        )UP**   0

       UP-PLANE  HIGH-RATE ZEROS

       (  -.9009481996     ,   0.               )
       (   .1830897924E-02,   0.               )
       (  -.5657507592     ,    6.861565379     )
       (  -.5657507592     ,   -6.861565379     )
       (  -4.956749255     ,   0.               )
       (  -9.892804162     ,   0.               )
       (  -14.00440188     ,   0.               )
       (  -15.45466095     ,   0.               )
       (   50.00000000     ,   0.               )

       UP-PLANE  HIGH-RATE POLES

       (  -.5087865665     ,    .3042290447     )
       (  -.5087865665     ,   -.3042290447     )
       (  -.1726986844E-02,   0.               )
       (  -2.169528987     ,    3.364297796     )
       (  -2.169528987     ,   -3.364297796     )
       (  -12.82548298     ,   0.               )
       (  -16.86376037     ,  -12.01331830      )
       (  -16.86376037     ,   12.01331830      )
       (  -10.52412126     ,   0.               )

       UP-PLANE  HIGH-RATE POLES

       (  -.5087865665     ,    .3042290447     )
       (  -.5087865665     ,   -.3042290447     )
       (  -.1726986844E-02,   0.               )
       (  -2.169528987     ,    3.364297796     )
       (  -2.169528987     ,   -3.364297796     )
       (  -12.82548298     ,   0.               )
       (  -16.86376037     ,  -12.01331830      )
       (  -16.86376037     ,   12.01331830      )
       (  -10.52412126     ,   0.               )
       (  -50.00000000     ,   0.               )

              MULTIPLICITY =  1
              MULTIPLICITY =  1
              MULTIPLICITY =  1
              MULTIPLICITY =  1
              MULTIPLICITY =  1
              MULTIPLICITY =  1
              MULTIPLICITY =  1
              MULTIPLICITY =  1
              MULTIPLICITY =  1
              MULTIPLICITY =  1
```

Figure 27a. TXCONV w'-Plane Output for Example 2

```
WP-PLANE  LOW-RATE NUMERATOR

(    1.000000000      )WP**  9
(    20.40510280      )WP**  8
(    872.5118006      )WP**  7
(   -2611.459068      )WP**  6
(   -114623.7735      )WP**  5
(   -1512307.530      )WP**  4
(   -9499854.210      )WP**  3
(   -26273236.55      )WP**  2
(   -17896690.98      )WP**  1
(    34333.56990      )WP**  0


WP-PLANE  LOW-RATE DENOMINATOR

(    1.000000000      )WP**  9
(    57.97720802      )WP**  8
(    1345.564154      )WP**  7
(    15812.11940      )WP**  6
(    101039.5972      )WP**  5
(    367374.2388      )WP**  4
(    777651.0181      )WP**  3
(    573346.2855      )WP**  2
(    105131.8811      )WP**  1
(    283.4745940      )WP**  0
```

```
         WP-PLANE  LOW-RATE ZEROS

(    .1013663807E-02,  0.                 )
(   -.0739169902    ,  0.                 )
(   -8.991321261    ,   8.877203838       )
(   -8.991321261    ,  -8.877203838       )
(   -10.48675667    ,   .7742549174       )
(   -10.48675667    ,  -.7742549174       )
(   -6.069304860    ,  -12.58225559       )
(   -6.069304860    ,   12.58225559       )
(    10.66666667    ,  0.                 )


         WP-PLANE  LOW-RATE POLES

(   -.5087966861    ,   .3040071080       )
(   -.5087966861    ,  -.3040071080       )
(   -.1726986839E-02,  0.                 )
(   -8.237556281    ,   3.352960724       )
(   -8.237556281    ,  -3.352960724       )
(   -10.94610689    ,  0.                 )
(   -10.05500688    ,  -4.673770650       )
(   -10.05500688    ,   4.673770650       )
(   -8.426654443    ,  0.                 )

LOW-RATE WP-PLANE  GAIN =   .8931044870E-02
HIGH-RATE SAMPLING PERIOD (SEC) =   .4000000000E-01
LOW-RATE  SAMPLING PERIOD (SEC) =   .1200000000
```

Figure 27b. TXCONV w'-Plane Output for Example 2

SECTION VI

SUMMARY

The high-speed timeshared digital computer with a large memory has made it possible for the analyst and the computing machine to be coupled closely together to solve a large variety of engineering problems. In system analysis and design, one of the essential ingredients for this coupling is a library of properly structured analysis and synthesis programs residing in the machine. Moreover, easily applied and accurate computer programs that deal with discrete or hybrid systems are becoming more essential as a result of the increasing use of digital controllers in automatic control systems. Practical hybrid systems contain continuous or analog elements (e.g., plant, process, or controlled element) which can be described by or approximated with linear differential equations and discrete elements (e.g., digital controller) which are inherently defined by difference equations. Thus, not only is the high-speed, large memory digital computer used to analyze and design discrete or hybrid systems, but the small-scale digital computer is becoming an integral part of the control system itself.

The fundamental first step in the analysis or design of a hybrid control system is to acquire or formulate a valid linear model of the system being considered. This includes the discretization of the continuous elements in the system into a valid discrete domain. The DISCRET computer program presented in this report provides this discretization. DISCRET takes a continuous element expressed as an s-plane transfer function and transforms it into the z-, w-, or w′-plane (see Eq. 58). The resulting discrete transfer function is exact as opposed to the approximate discrete transfer function obtained from a substitution-for-s approach such as the Tustin or first-difference transforms. The discrete transfer function generated by DISCRET defines the continuous variables (associated with the continuous element) at each sampling instant of the sampler device in the system. It is assumed that the

sampler passes the continuous signal at discrete instances equally spaced in time. The time interval T between these samples is called the sampling period. Once calculated, this discrete model of the continuous element can be readily combined with the inherent discrete elements in the system (such as a digital controller) to provide a unified, concise description of the total system. An inherently discrete element in a hybrid system is first modeled with a recursion or difference equation and then directly converted to the z-, w-, or w′-plane by substituting the z^{-n} delay operator for each discrete term in the difference equation. Therefore, all the elements in a hybrid system can be described by transfer functions in the z-, w-, or w′-plane. Consequently, similar frequency and time domain techniques used for continuous s-plane systems can be applied by the control system engineer to a wide variety of practical hybrid systems.

Many practical hybrid control systems are also multi-rate in nature. That is, they contain samplers which operate at different sample rates. This adds additional complexity to the analysis and design procedures, but concise, definitive methods are available for handling multi-rate systems (e.g., Refs. 1-3). However, a troublesome situation arises when the output of a multi-rate system is sampled at a lower rate than its input. This sampling configuration invariably leads to the requirement of computing a low-rate discrete transform from a given high-rate discrete transform (see Eqs. 76-78). The TXCONV computer program presented in this report calculates this complex transformation and makes it a routine operation in the analysis and design process. Specifically, TXCONV takes a high-rate discrete transfer function expressed in the z-, w-, or w′-plane and transforms it into a desired low-rate discrete transfer function in the z-, w-, or w′-plane. This relieves the analyst from the computationally involved procedure of calculating the residues of a complex integral.

Both the DISCRET and TXCONV computer programs represent essential tools for dealing with multi-rate, hybrid control systems. They provide two of the primary techniques used to formulate a unified discrete model of a complex, hybrid system. However, they are by no means the only

tools or techniques required by the analyst. Other computational operations that are required include polynomial and transfer function manipulation routines. These routines are needed to handle the individual discrete transfer functions present in single-loop and multiloop hybrid systems. Analysis routines that calculate root locus, frequency response, and time response in the discrete domain for single-rate and multi-rate, hybrid systems are also needed. Most of these routines for single-rate discrete or hybrid systems are already available in the TOTAL computer program (Ref. 13). Additional software routines that specifically address multi-rate systems and the special features involved in calculating the continuous frequency response of a discretely excited system (Ref. 1) are presently under development at AFWAL/FIGC.

Both DISCRET and TXCONV have been integrated into the total computer program. DISCRET and TXCONV are two of the over 100 options available during the interactive execution of the TOTAL program. This interactive feature allows a close coupling between the analyst and the computing machine (digital computer) such that a real-time dialog between the two can be effectively carried out. This results in a more effective and efficient usage of the computing machine and improves the accuracy and speed of the analysis and synthesis process.

REFERENCES

1. Whitbeck, R. F., and L. G. Hofmann, Analysis of Digital Flight Control Systems with Flying Qualities Applications, AFFDL-TR-78-115, Sept. 1978.

2. Ragazzini, J. R., and G. F. Franklin, Sampled-Data Control Systems, New York, McGraw-Hill, 1958.

3. Tou, J. T., Digital and Sampled-Data Control Systems, New York, McGraw-Hill, 1959.

4. Saucedo, R., and E. E. Schiring, Introduction to Continuous and Digital Control Systems, New York, Macmillan, 1968.

5. Jury, E. I., Sampled-Data Control Systems, New York, Wiley, 1958.

6. Jury, E. I., Theory and Application of the z-Transform Method, New York, Wiley, 1964.

7. Kuo, B. C., Analysis and Synthesis of Sampled-Data Control Systems, Englewood Cliffs, N. J., Prentice-Hall, 1963.

8. Lindorff, D. P., Theory of Sampled-Data Control Systems, New York, Wiley, 1965.

9. Tsypkin, YA. Z., Sampling Systems Theory and Its Application, Vols. 1 and 2, [tr. by R. C. Hutchison], New York, Macmillan, 1964 (original Russian publication, Moscow, 1958).

10. Oldenbourg, R. C., and H. Sartorius, The Dynamics of Automatic Controls, [tr. by H. L. Mason], New York, American Society of Mechanical Engineers, 1948.

11. Wilts, C. H., Principles of Feedback Control, Reading, Mass., Addison-Wesley, 1960 (Appendix B, p. 261).

12. Herbert, E. E., and B. Rogerson, Digital Computer Program for the Determination of Modified z-Transforms with w-Plane Transformation, U.S. Army Missile Command, Redstone Arsenal, Ala., Tech. Rept. RG-75-20, Dec. 1974.

13. Larimer, S. J., An Interactive Computer-Aided Design Program for Digital and Continuous Control System Analysis and Synthesis (TOTAL), AF Institute of Technology, Wright-Patterson AFB, M.S. Thesis, Mar. 1978 (AD A055 418).

14. Fortran Extended Version 4 Reference Manual, Control Data Corporation, Sunnyvale, Calif., Pub. No. 60497800, Rev. D, 1978.

☆U.S.Government Printing Office: 1981 — 757-002/436